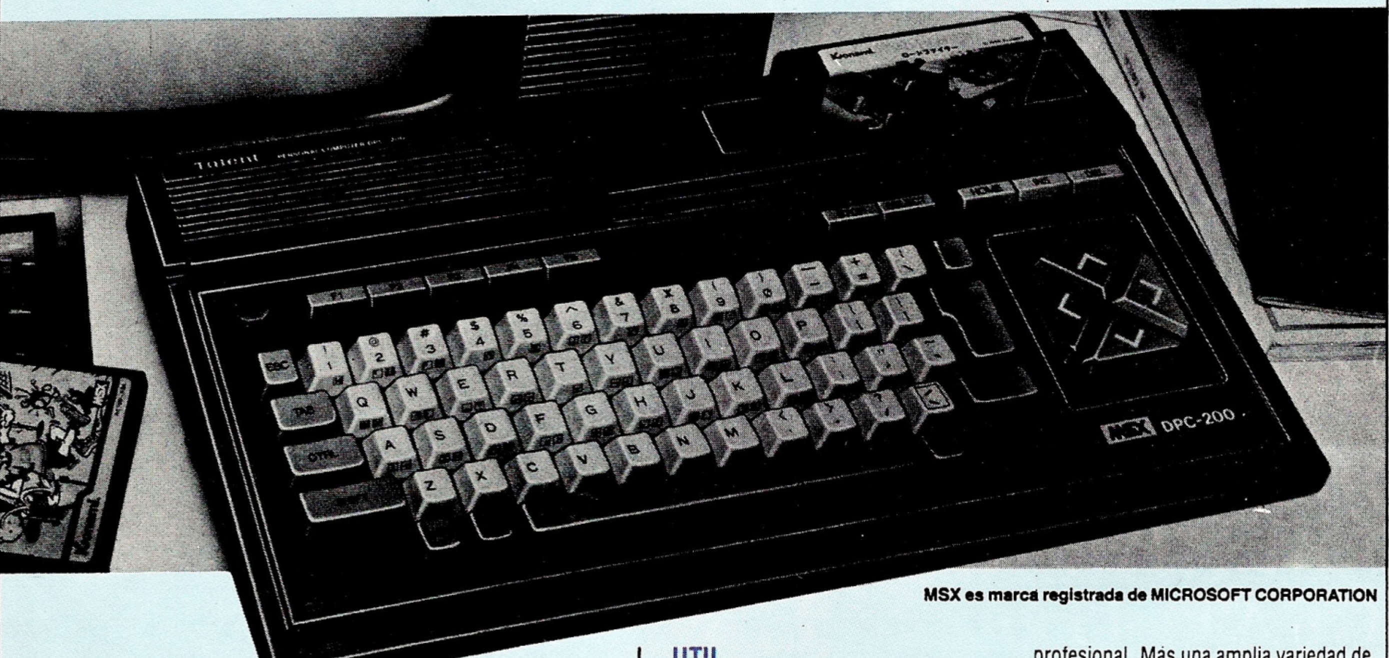

SEPTIEMBRE 1988
REVISTA PARA USUARIOS DE LA NORMA
LOAD MSX
AÑO 3 Nº 29
A28 REP.ARGENTINA
COMO USAR UNA BASE DE DATOS
DISEÑO DE CARACTERES
• CORREO ELECTRONICO GRATUITO
• FACIL SOFT PARA HACER SPRITES



Talent
Konami
DPC-200

STAFF

LOAD MSX

Director Periodístico

Fernando Flores

Secretario de Redacción

Ariel Testori

Redacción

Fernando Pedró

Arte y Diagramación

Fernando Amengual

Departamento de Publicidad

Dolores Urien

Servicios Fotográficos

Image Bank, Oscar Burriel y Victor Grubicy

EDITORIAL PROEDI

Presidente

Ernesto del Castillo

Vicepresidente

Cristian Pusso

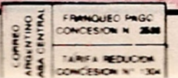
Director Titular

Javier Campos Malbrán

Director Suplente

Armengol Torres Sabaté

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Registro Nacional de la Propiedad Intelectual: E.T.M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual. Todos los derechos reservados. ISSN 0326-8241. Impreso en Impresiones Gráficas Tabaré S.A.I.C., Erézcano 3158, Capital. Fotocromo de tapa: Columbia. Los ejemplares atrasados se venderán al precio del último número en circulación. Prohibida la reproducción total o parcial de los materiales publicados, por cualquier medio de reproducción gráfico, auditivo o mecánico, sin autorización expresa de los editores. Las menciones se realizan con fines informativos y técnicos, sin cargo alguno para las empresas que comercializan y/o los representan. Al ser informativa su misión, la revista no se responsabiliza por cualquier problema que pueda plantear la fabricación, el funcionamiento y/o aplicación de los sistemas y los dispositivos descriptos. La responsabilidad de los artículos firmados corresponde exclusivamente a sus autores. Distribuidor en Capital: Martino, Juan de Garay 358, P.B., Capital. Distribuidor en el interior: D.G.P., Hipólito Yrigoyen 1450, Capital, Tel.: 38-9266/9800.



SUMARIO

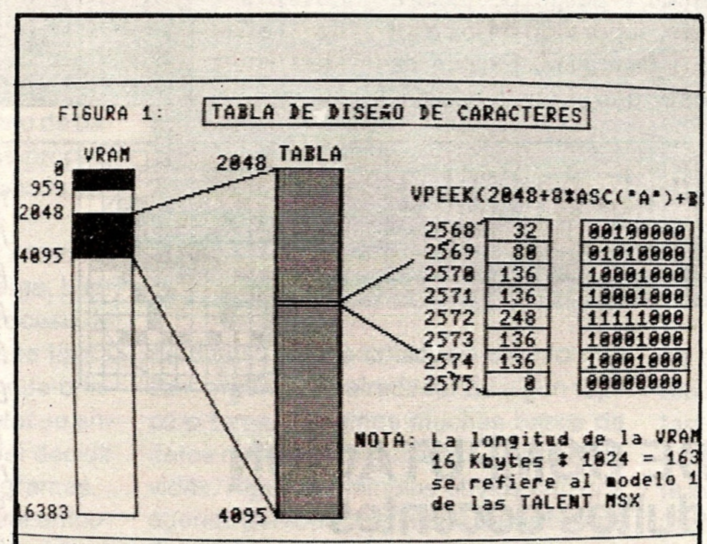
AÑO 3 Nº 29

COMO SE USA UN SISTEMA DE BASE DE DATOS



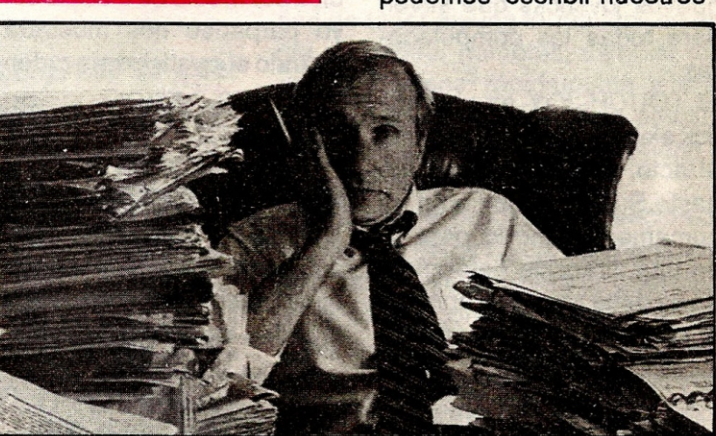
Al principio no pudimos esperar para comenzar a utilizar nuestro sistema de base de datos... y ahora, con el disquete casi lleno, nos damos cuenta de que el formato armado con tanto trabajo sencillamente no sirve... Trataremos de evitar esta clase de problemas aclarando algunos puntos importantes en el uso de esta importante herramienta. (Pág. 5)

DISEÑO DE CARACTERES



Mostramos cómo una parte de la memoria almacena los caracteres. Su comprensión nos servirá cuando necesitemos diseñar alguna letra o símbolo en particular. (Pág. 7)

RINCON DEL USUARIO



Se presenta un utilitario que permite leer archivos generados por el dBASE II desde el MSX BASIC y generar un programita de ejemplo que arma los campos del archivo. (Pág. 20)

CORREO ELECTRONICO GRATUITO

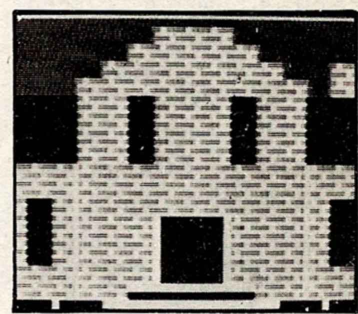
Explicamos cómo enviar un archivo previamente almacenado en ACAmática utilizando un modem Talent TMX-510. (Pág. 24)

POSIBILIDADES DEL PROCESADOR DE TEXTOS

Les mostramos cómo podemos escribir nuestros programas en el MSX-WRITE. (Pág. 30)

PROGRAMAS

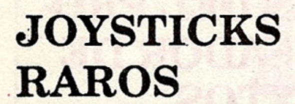


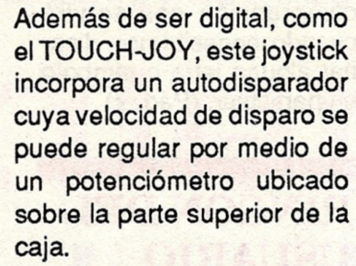


Sprites (Pág. 10) - El juego de los detectives (Pág. 14)

SECCIONES FIJAS

Noticias MSX (Pág. 4) - Soft al día (Pág. 32) - Buzón (Pág. 34)





JOYSTICKS RAROS

La empresa I.S.F.P. acaba de lanzar al mercado local dos nuevos joysticks que sirven para todas las computadoras.

El TOUCH-JOY es un joystick electrónico, que funciona al tacto, y por lo tanto sin desgaste.Su funcionamiento es sumamente sencillo, para "ir" en una dirección determinada se debe tocar (no apretar) el contacto correspondiente a esa dirección. No más resortes, palancas ni microswitches que se puedan romper.

En la parte superior del josytick, hay un LED que nos indica que el mismo está encendido y listo para funcionar.

El otro joystick es el OMNI-FIRE, con una innovación que puede llegar a resultar muy divertida. Su accionamiento es por la voz humana.

Además de ser digital, como el TOUCH-JOY, este joystick incorpora un autodisparador cuya velocidad de disparo se puede regular por medio de un potenciómetro ubicado sobre la parte superior de la caja.

La velocidad de disparo se indica por medio de un LED cuyo parpadeo nos mostrará cuándo el joystick da la orden de disparo.

VIRUS

Uno de los males que están sufriendo los sistemas de computación complejos de grandes empresas o instituciones públicas son los virus.

Estos son programas que se introducen en las grandes redes con el fin de destruir o alterar información, por lo general secreta o reservada.

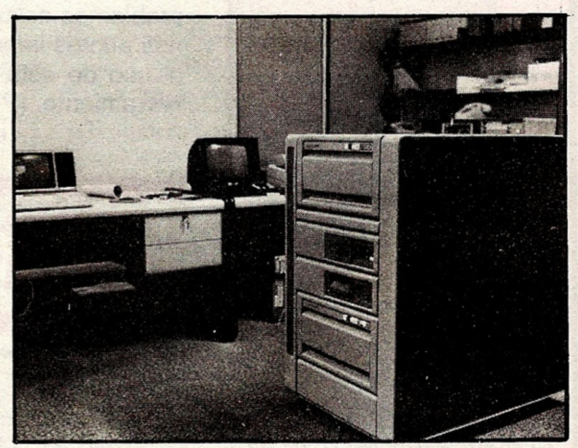
Los virus han sido introducidos en instituciones como IBM, la NASA y el Providence Journal-Bulletin.

Si bien las empresas dicen que los virus de las computadoras no son males importantes, el peligro potencial existe, sobre todo en los espías industriales. Existen casos en que un espía industrial se infiltró en los sistemas de computación de compañías competidoras, para colocar programas que destruyen parte de la información almacenada o producen errores constantes en la facturación o descienden la calidad del producto que se fabrica.

NUEVO CONCURSO



El certámen "Una computadora para mi escuela", organizado por el programa de ATC **Historias de la Argentina Secreta** con el auspicio de la revista **K-64** fue un éxito. Prueba de ello es el número importante de concursantes que participaron con el objeto de ganar un equipo completo de Talent MSX para su escuela.



Ahora, debido al eco obtenido, **Historias...** vuelve a la carga con un nuevo certámen.

Esta vez se trata de **El viaje de Historias de la Argentina Secreta.**

Consiste en una selección entre quienes hayan participado del concurso "El Programador del Ano" para formar un equipo que diseñará un juego educativo sobre la base de la temática del programa de T.V. y radio.

DELPHI

Los usuarios de Delphi Argentina podrán usar el correo electrónico, vía Mail, desde los Estados Unidos. También usando Delphi USA tendrán acceso directo a Delphi de nuestro país. Si un usuario quisiera enviar un mensaje a algún otro deberá usar la opción correspondiente, dejando constancia del nombre del destinatario.

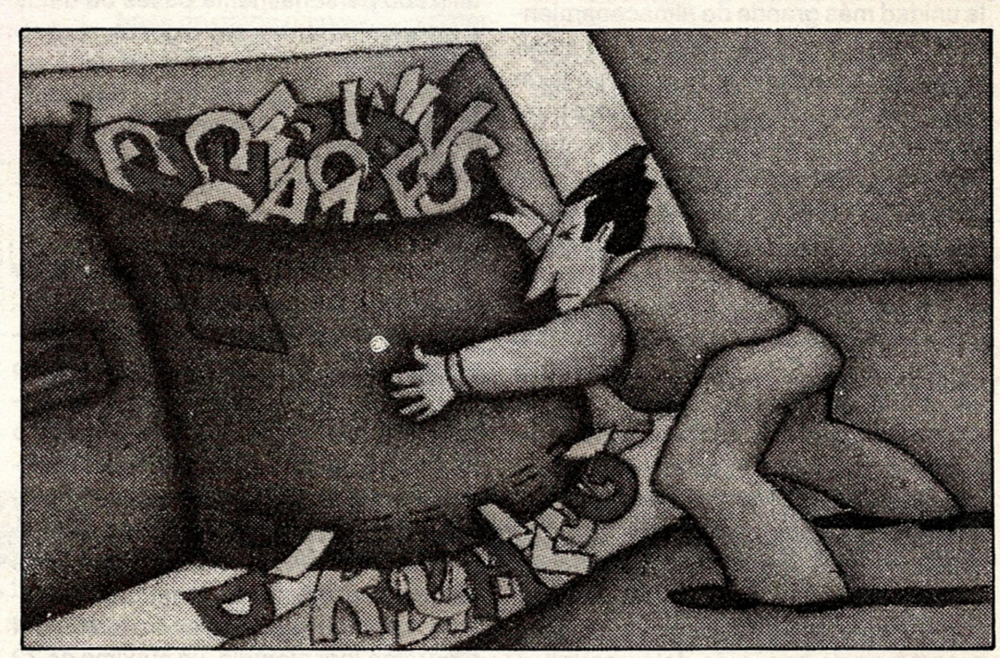
COMO SE USA UN SISTEMA DE BASE DE DATOS

Al principio no pudimos esperar para comenzar a utilizar nuestro sistema de base de datos... y ahora, con el disquete casi lleno, nos damos cuenta de que el formato armado con tanto trabajo sencillamente no sirve... Trataremos de evitar esta clase de problemas aclarando algunos puntos importantes en el uso de esta importante Herramienta.

Si somos usuarios que utilizamos la microcomputadora como una ayuda en el manejo de las finanzas hogareñas, clubs o pequeños negocios (lo que generalmente denominamos "uso serio") generalmente necesitaremos tres programas para manejar todas las rutinarias tareas manuales: un procesador de texto, una planilla electrónica y un programa de almacenamiento y recuperación de información (conocidos en el círculo de computación como un "sistema de manejo de base de datos").
Se ha escrito bastante sobre el uso apropiado de los procesadores de texto y las planillas de cálculo, mientras que los principiantes en bases de datos no tienen tanta bibliografía. Excepto por un mínimo "know how" (saber cómo) que viene con el manual del programa, el usuario de base de datos queda librado a sus propias fuerzas, rodeado de términos pocos conocidos y comandos confusos.
Este artículo brinda una breve introducción a las bases de datos y su jerga. Luego avanza paso a paso en el proceso de resolver un problema usando una típica base de datos. La información que brindaremos servirá para que el lector se encuentre más "armado" para poder decidir la compra de uno de estos programas.
La información contenida en este artículo se aplica a cualquier programa de base de datos y a cualquier computadora (pero desde la óptica MSX, naturalmente).

LAS BASES DE LAS BASES DE DATOS...

¿Qué es un programa de base de datos y para qué puede servirnos? Una "base



de datos" es una colección de información organizada alrededor de algún tópico o tema. Tenemos muchas bases de datos no computadorizadas en nuestras vidas. Algunos ejemplos obvios son una agenda personal, un listado de los bienes de una casa para el seguro o una lista con las tarjetas que se deben enviar en Navidad. Un ejemplo de un producto de una base de datos computadorizada es la guía telefónica.
Si estamos lidiando con un volumen importante de información, o si la información cambia con bastante frecuencia, nos podremos beneficiar usando un programa de base de datos.
Cualquiera que haya intentado mantener un listado de correos -"mailing list"- o (¿se imaginan?) llevar a mano la guía telefónica puede testificar sobre la necesidad de tener alguna forma fácil de agregar y remover nombres, cambiar direcciones y ordenar rápidamente la lista en orden alfabético o por código postal. Un programa de base de datos bueno nos permitirá hacer todo esto.
Para ilustrar el proceso de creación de una base de datos típica, diseñaremos un inventario hipotético para un gran club de usuarios de computadoras... MSX. En nuestro club imaginario, compramos algunos productos en liquidaciones u obtenemos programas de "dominio público" y luego los prestamos a nuestros miembros. Nuestro ejemplo sólo sugiere lo que se puede hacer con este tipo de material, pero no es un tratamiento exhaustivo de la situación.

TERMINOLOGIA

Toda área del conocimiento humano tiene sus propias palabras que nos permiten hablar específicamente de lo que estamos haciendo. La computación no es la excepción. Si queremos aplicar realmente lo que estamos aprendiendo en un programa de base de datos verdadero, deberemos familiarizarnos con los pocos términos que describiremos a continuación.
Si por el trabajo llevamos registros (fi-

chas) de algún tipo, probablemente estemos familiarizados con los objetos físicos utilizados.
La información puede mantenerse en un formulario que consiste de una o varias páginas. El formulario deberá incluir un número de preguntas, cada una seguida por un espacio en blanco en donde podemos escribir la respuesta. El espacio en blanco puede ser corto si está previsto para contenter un código postal. Podría ser muy largo, si está destinado a almacenar un nombre de calle. Todos los formularios del mismo tipo pueden guardarse en una carpeta de archivo. O, si realmente son muchos, se pueden almacenar en dos o más carpetas.
De tiempo en tiempo, podríamos desear recorrer todos los formularios y resumir la información contenida en ellos. También podríamos desear sacar una lista de todos los formularios de cierto tipo en orden alfabético y sumar varios tipos de información, proveyendo los totales al final de los resúmenes. Este tipo de resumen es el producto principal derivado de un ordenamiento de formularios dado, y se denomina "reporte" (report).
Un sistema de ficheros electrónico es similar, pero utiliza su propio lenguaje especial. Tenemos el "archivo" (file) como la unidad más grande de almacenamiento de datos en nuestro sistema. Un archivo de datos contiene toda la información de un individuo que la computadora puede manejar a la vez. Por lo tanto, nuestro club de usuarios podría tener un archivo con su inventario y otro para el listado de tarjetas de Navidad. Si nuestro archivo es muy grande, podríamos tenerlos en dos o más disquetes de datos.
Los archivos de datos están compuestos por "registros". Si fuéramos la secretaria de un club de usuarios grande y tuviéramos un archivo que contiene toda la información de todos nuestros socios, entonces todo lo que necesitaríamos saber sobre un socio sería un solo registro. Un registro es el equivalente lógico de un formulario. Así como un formulario puede componerse de varias páginas, algunos programas de bases de datos permiten tener información de más de una página de pantalla en un registro. Una página de pantalla consiste en toda la información que se puede ver en la pantalla de la computadora simultáneamente.
Y así como un formulario está compuesto de diversas preguntas y espacios en blanco para rellenar, un registro está compuesto de "campos" (fields). Un campo es un ítem de información de un registro. Los campos típicos de un registro de un miembro pueden ser su apellido, número telefónico o código postal. Un ítem de un formulario puede tener lugar como para una respuesta amplia o pequeña. Un campo trabaja de la misma forma.
De hecho, la persona que inicializa una base de datos debe especificar el número máximo de caracteres que puede existir en cada campo. La unidad utilizada para medir la "longitud de campo" es el "carácter". Un carácter es una letra, número o espacio. Un número telefónico podría tener 14 caracteres de longitud: (0320)628-8752.

PASOS EN EL DISEÑO

Siempre debemos seguir 10 pasos básicos al crear un sistema de mantenimiento de datos en una computadora. Estos pasos se aplican también si se utiliza una base de datos comercial (un programa como el IdeaBase, el fichero de Bitgame o el dBASE II). Si se siguen con cuidado, estos pasos asegurarán que nuestro sistema nos brindará los resultados que deseamos y necesitamos. A pesar de haber utilizado personalmente bases de datos e incluso haber escrito algunos programas de base de datos por años, debo seguir todavía los mismos pasos de diseño que listamos a continuación. No hay ninguna clase de atajos.

1. LEAMOS EL MANUAL

A pesar de que el primer paso debería ser obvio, rara vez se cumple.
"Lea el manual primero y realice los ejercicios religiosamente". No hay mucho que preguntar cuando se está aprendiendo a usar un programa nuevo, pero mucha gente comienza su periplo insertando el disco, encendiendo la computadora e intentando inicializar un archivo.
Es aconsejable usar un block de apuntes mientras se trabaja con el manual. Tomemos nota sobre cualquier limitación del programa (por ejemplo, un máximo de 32 campos por registro) o problemas. Tratemos de encontrar las respuestas a las preguntas con el vendedor que nos proveyó el soft (o recurra a la Hot-Line - 38-6601, si el producto es de Talent). Después de todo, si tenemos problemas haciendo los ejemplos del manual, imaginemos los que podríamos tener si tratamos de resolver un problema real con el programa.

2. EFECTUEMOS EL TRABAJO DE DISEÑO EN EL PAPEL

"¿Para qué molestarnos en escribir todo en papel? ¿Para qué tenemos una computadora?" El mayor problema en cualquier área de la computación "seria" es la tendencia a saltar a la mitad del proceso antes de tomarse tiempo para trabajar en papel siquiera lo que deseamos hacer. Si poseemos un procesador de texto, podemos utilizarlo en el trabajo de diseño sobre papel. No comencemos a trabajar en la inicialización de la base de datos hasta haber cumplido todos los pasos hasta el noveno. Si no los cumplimos, seguramente tendremos problemas.

3. DETERMINEMOS LOS RESULTADOS DESEADOS Y TRABAJEMOS "REMONTANDO LA CORRIENTE"

Lo más natural es hacer una lista de los campos que deseamos incluir en la base de datos y luego setearlos. En este caso, lo más natural es lo incorrecto. Si encaramos el problema de esta forma, lo más probable es que tengamos un campo o dos que nunca necesitemos y nos olvidemos de algún campo vital de cierto reporte.
Estamos usando un programa de administración de base de datos porque queremos obtener alguna clase de resultado, generalmente impreso en papel, un reporte de algún tipo.
Si comenzamos con una lista de reportes diseñada cuidadosamente y trabajamos "remontando la corriente" hacia los campos necesario para obtener ese resultado, seguramente ahorraremos espacio de máquina valioso eliminado en el archivo con información innecesaria.
Si necesitamos duplicar un reporte manual existente, las opciones son simples. Si es un reporte nuevo o queremos realizar cambios en un sistema existente, podríamos desear sentarnos frente a una máquina de escribir (o procesador de texto) y escribir el reporte con datos de ejemplo (Ver la Figura 1).
Una vez que tenemos el reporte de ejemplo en la mano, el resto del proceso se ejecutará con facilidad. Si también planeamos que los datos se vean en pantalla, podemos crear pantallas de reporte de ejemplo en papel, tomando en consideración el tamaño y capacidad de la pantalla de la computadora (por ejemplo, 40x24 caracteres en MSX1 u 80x24 caracteres en MSX2).

4. HAGAMOS UNA LISTA DE CAMPOS

Si no se utiliza en la pantalla o en un reporte impreso, no lo necesitamos. Escri-

Figura 1

Reportes sobre Software y Equipamiento (Ejemplos)

Inventario (Reporte 1)

Nombre Producto	Número Inventario	Costo Unit.	Precio Unit.	En Stock	En Préstamo

Lista de Precios (Reporte 2)

Nombre Producto	Número Inventario	Precio Unitario

Compras propuestas (Reporte 3)

Nombre Producto	Número Inventario	Necesidad	Costo Unitario	Total

Valor de Inventario (Reporte 4)

Nombre Producto	Número Inventario	En Stock	Costo Unitario	Valor

Figura 2

Campos Ingresados	Campos calculados
Nombre Producto	Compra propuesta
Número Inventario	Total
Costo	Valor
Precio	
En stock	
En préstamo	
Items máximos en stock (no se imprime, pero se usa en el Reporte Compra Propuesta)	

Figura 3

Fórmulas utilizadas en los campos calculados

NECESIDAD	= MAXIMO menos EN STOCK menos EN PRESTAMO
TOTAL	= COMPRA PROPUESTA veces COSTO (CP * COSTO)
VALOR	= EN STOCK veces COSTO

bamos cada ítem en cada reporte. Si es un dato que se necesita ingresar, pongámoslo en una lista. Si es un dato que puede ser calculado por el programa, pongámoslo en otra lista. (Ver Figura 2).

Ahora observemos la lista de los campos calculados por la computadora. Escribamos la información que necesitará la misma para poder calcular estos ítems. ¿Está incluida en la lista de campos? Si no lo está, agreguémosla. Ahora escribamos cualquier fórmula necesaria (PRECIO = STOCK veces COSTO UNITARIO) (Ver la Figura 3). ¿Nuestro sistema de base de datos puede efectuar este tipo de operaciones? Si no estamos seguros, fijémonos en el manual y verifiquémoslo.

5. HAGAMOS UNA LISTA CON LAS CARACTERISTICAS DE LOS CAMPOS

La mayoría de los programas de base de datos quieren que determinemos como mínimo dos cosas sobre cada campo: una es la longitud máxima del mismo. Algunas longitudes de campo son fáciles de determinar. Un campo que contiene el código postal tendrá una longitud de 4 caracteres y un teléfono local, de ocho.

Pero, ¿qué sucede con un apellido o una dirección postal? Estos campos requerirán una pequeña investigación estadística. Si se especifica una longitud demasiado corta, deberemos usar abreviaturas para los datos de mayor longitud. Y poca gente ve con agrado que se abrevie su apellido.

Y si esa longitud es excesiva, se desperdiciará espacio de almacenamiento y es posible que no quepa en el formulario impreso o etiqueta de mailing. Tratemos de buscar en los registros para averiguar qué longitud tienen los ejemplos más extensos. También tengamos en cuenta las limitaciones de impresión de la impresora o del formulario de impresión (por ejemplo, si usamos una etiqueta de 6 cm, ninguna línea de dirección postal podrá exceder un total de 35 caracteres en letra normal de impresora).

El otro ítem que se debe determinar es si el campo estará diseñado para aceptar datos numéricos (campo numérico) o si un campo puede contener tanto letras como números (campo alfanumérico o de carácter, a veces denominado "string").

Como una regla general, todos los campos deben setearse como alfanuméricos a menos que necesitemos utilizar matemáticas con ellos. Por el sólo hecho de contener números, esto no hace que automáticamente deba ser un campo numérico. Es muy difícil que necesitemos multiplicar un número telefónico por un código postal. Algunos programas predefinen los tipos de campos de acuerdo al tipo de datos que éstos van a albergar (como el signo $ o la fecha).

Figura 4

Campos	Tipo Campo	Longitud
Nombre Producto	Alfa	20 <- Clave primaria
Presentacion	Alfa	10
Número de orden	Alfa	15 <- Clave secundaria
Costo Unit.	Numérico	5
Precio Unit.	Numérico	5
En stock	Numérico	5
En préstamo	Numérico	5
Items máximos en stock	Numérico	2
Calculados:		
Necesidad (Max-En stock-En préstamo)	Numérico	2
Total (Necesidad * Costo)		6
Valor (En stock * Costo)		6

6. LISTEMOS LOS CAMPOS EN UN ORDEN LOGICO

El truco en el diseño de un formulario es armarlo en el orden más natural posible. El siguiente formulario sería muy difícil de completar:

Ciudad...
Apellido...
Código Postal.....................................
Calle...
Nombre...
Provincia..

Es mucho más probable que la información nos llegue en el orden: Nombre, Apellido, Calle, Código Postal, Ciudad, Provincia. Cualquier variación en este orden confundiría a la persona que debe ingresar los datos. Lo mismo puede decirse de cualquier bloque de datos, en mayor o menor grado. Algunos arreglos serán más claros y más naturales que otros. Re-arreglemos los campos en un orden que pensemos más lógico y fácil de usar.

7. DETERMINEMOS LOS CAMPOS PARA ORDENAR

Una de las tareas más útiles que se puede ejecutar en cualquier programa de base de datos es arreglar la información en orden específico, comúnmente denominado "sort". Podríamos necesitar imprimir etiquetas postales ordenadas por código postal para el correo. O también podríamos tener la misma información ordenada por apellido. Se pueden encontrar muchas formas de arreglar los datos, cuando la computadora hace todo el trabajo para uno.

Algunos programas de base de datos requieren que se determinen los campos que utilizaremos al diseñar el archivo. Si es así, pensemos detenidamente qué campo utilizaremos para el ordenamiento. Los programas de este tipo "se achanchan" terriblemente cuando existen numerosas claves de ordenamiento, que también devoran rápidamente el espacio del disquete (Ver la Figura 4 para el armado final de campos).

8. DISCUTAMOS LOS RESULTADOS CON OTRAS PERSONAS

Por aquello de que "cuatro ojos ven más que dos", siempre otros podrán ver detalles que no vimos y necesidades que olvidamos considerar, por más experto que nos consideremos. Por lo tanto, aunque es bueno involucrar a todas las personas que necesitan la información que estamos almacenando en cada paso del diseño, es imperativo que otros también "se metan".

El próximo paso es inicializar el archivo en la computadora. Las cosas que sólo son ideas en este momento pueden ser muy difíciles de cambiar una vez que la información sea ingresada en la computadora. Una hora en este momento puede ahorrar días más adelante.

9. INICIALIZAR EL ARCHIVO

Por fin podemos encender la computadora. Ahora es tiempo de revisar los pasos que nos brinda el manual e inicializar el archivo y los reportes. Si hemos completado los ocho pasos anteriores y tenemos a mano todas las notas, este paso debería ser fácil.

10. PROBEMOSLO

Hagamos una copia del disco de datos e ingresemos una docena de registros. Luego probemos todos los reportes y verifiquemos que muestren lo que deseamos. Si cometimos un error grueso de diseño, no nos preocupemos por comenzar de nuevo. Es más fácil rediseñar el archivo de datos en este punto que cuando ya tenemos una gran cantidad de datos almacenados y tenemos que rehacer todo más tarde.

Si trabajamos bien, continuemos ingresando el resto de los datos. Guardemos una copia del disco de datos cuando no tengamos aún ninguno ingresado. De esta forma, cada vez que necesitemos otra copia de esta base de datos, por ejemplo con datos del año siguiente, bastará con copiar este disco de datos en blanco y usarlo.

UNA APLICACION PERSONAL

Ahora que hemos seguido el proceso en la teoría, podríamos tomarnos un poco de tiempo y aplicarlo a un problema particular. Tenemos que llevar cuenta de muchas cosas, pero veamos algo relacionado con computadoras.

Ya que no toma mucho tiempo llenar nuestros escritorios con una cantidad "enorme" de discos (o cintas), muchos con una gran cantidad de programas, es muy fácil perder la noción de qué hay y en qué disco.

Si trabajamos con una MSX podremos usar alguna de las bases de datos disponibles: dBASE II, IdeaBASE, MS BASE, CardBox, Citation (para MSX2), etcétera.

A pesar de usar un programa simple de base de datos, no encontramos ninguna limitación que pueda causar problemas para esta aplicación. Si usamos el IdeaBASE, la ficha puede tener tantos campos y caracteres como entran en la pantalla; más que suficiente para nuestro caso.

Llevamos nuestro trabajo prolijamente organizado en papel, listando en los siguientes pasos las cosas que necesitaremos incluir en la base de datos y las decisiones que tomamos.

Queremos tres productos diferentes de nuestra base de datos. Primero, un listado alfabético de nuestros programas y el número de disco en que se encuentran. Previamente, decidimos numerar los discos secuencialmente y almacenarlos en cajas con un rótulo donde se indica claramente el rango de discos almacenados en ellas.

Segundo, queremos un listado de nuestros programas por categoría. Si queremos usar un programa educativo, podemos buscar qué opciones tenemos y dónde encontrarlas.

Tercero, podríamos valuar nuestro soft-

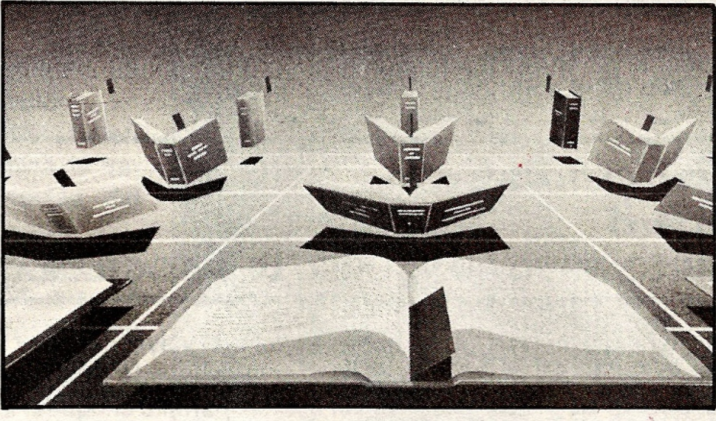
ware para poder asegurarlo. Muchos programas fueron comprados y sería caro reemplazarlos. Algunos son software de "dominio público", que se distribuyen gratis en nuestro club de usuarios (o por poco dinero).
Los reportes tendrían el siguiente formato:

POR NOMBRE
Nombre Programa: Número Disco: Comentarios:

POR CATEGORIA
Categoría: Sub-Categoría: Nombre Programa: Número Disco: Comentarios

REPORTE COSTO
Nombre Programa: Costo:

Ahora veamos nuestros reportes y hagamos una lista de los campos utilizados.



Quedaría algo así:
Nombre Programa
Número Disco
Comentarios
Categoría
Sub-Categoría
Costo
Los pasos finales son:
Listar las características de los campos.
Arreglar los campos en un orden lógico.
Elegir las claves de ordenamiento.
Estas tres opciones se reflejan en el listado de campos (Figura 5)
Ahora podemos charlar con la gente que compartirá el uso de nuestro sistema. Si se ve bien, podemos ir inicializando nuestra base de datos y verificarla con algunos datos reales. La figura 6 muestra parte de un reporte.

CONCLUSION

La administración de una base de datos es una de las tareas más útiles de las computadoras. Su potencial queda limitado únicamente por la imaginación del usuario y la relativa falta de familiaridad en el proceso.
Lamentablemente, la forma correcta de utilizar un programa de base de datos no es obvia. El usuario necesita pensar todo el proceso con cuidado antes de ir a la computadora. El tiempo que se invierta trabajando en papel se verá recompensado con creces en la facilidad con que se creará la base de datos y en su utilidad.
Podremos verificar que al encarar la resolución de problemas en forma sistemática con nuestra computadora siempre implicará una mejora en el producto obtenido y menores pérdidas de tiempo. El proceso aquí descripto ha sido probado y verificado reiteradamente, y garantiza el aprovechamiento total de nuestra base de datos.

Figura 5

Campo	Tipo	Longitud	Clave para ordenar
Número Disco	Numérico	3	No
Nombre programa	Alfa	20	Sí
Categoría	Alfa	15	Sí
Sub-categoría	Alfa	15	Sí
Costo	Numérico	5	No
Comentarios	Alfa	40	No

Figura 6

Archivo: MIS DISCOS
Reporte: Ejemplo 5

Número Disco	NOMBRE PROGRAMA	CATEGORIA	SUB-CATEGORIA	COSTO	COMENTARIOS
2	Planet Radar	Juego	Naves	0	Muy lento
2	Hammurabi	Juego	Aventuras	0	
5	Invierte y gana	Juego	Educativo	51	Entretenido
3	Dcopy	Utilitario	Programación	0	Diskcopy
4	MSX Utilities	Utilitario	Programación	120	Tipo Norton!
1	Extensiones Logo	Utilitario	Programación	320	Excelente

SPRITES

Autor: Diego Hernán Conde

Tipo: Utilitario

PARTICIPA DEL CUARTO CONCURSO DE PROGRAMAS

La necesidad de tener un método sencillo de crear SPRITES sin tener que recurrir a las hojas cuadriculadas y a la engorrosa tarea de transcribir los valores correspondientes en decimal o hexadecimal fue la base para la creación de este programa, de mucha utilidad para los programadores de juegos.

Desde la presentación hasta la salida por impresora este programa ofrece diversos aspectos destacables.

La portada del juego es superior a muchas de juegos comerciales, hecho que se repite en la mayoría de los participantes del 4to. Concurso. En ella se puede destacar la rutina de rotación de sprites.

El seguimiento del programa es muy simple, ya que se maneja mediante menúes, y tiene la posibilidad de pedir información para saber qué orden se debe seguir para llegar al objetivo propuesto.

FORMA DE USO

La creación del sprite se realiza mediante la pulsación de:

a) la barra espaciadora, para que el cursor (mano) cambie de trazo. Es decir que si estaba escribiendo pase a borrar, y viceversa.

b) las teclas del cursor en todas sus direcciones.

c) RETURN para ver el menú principal o regresar al cuadro de trabajo.

d) los números del 1 al 9.

MENUES

1.CUADRO: "blanco" o "negro", pinta el cuadro de uno de esos colores y queda listo para comenzar a definir un nuevo sprite. "Manejo del cursor" permite controlar la velocidad de la mano.

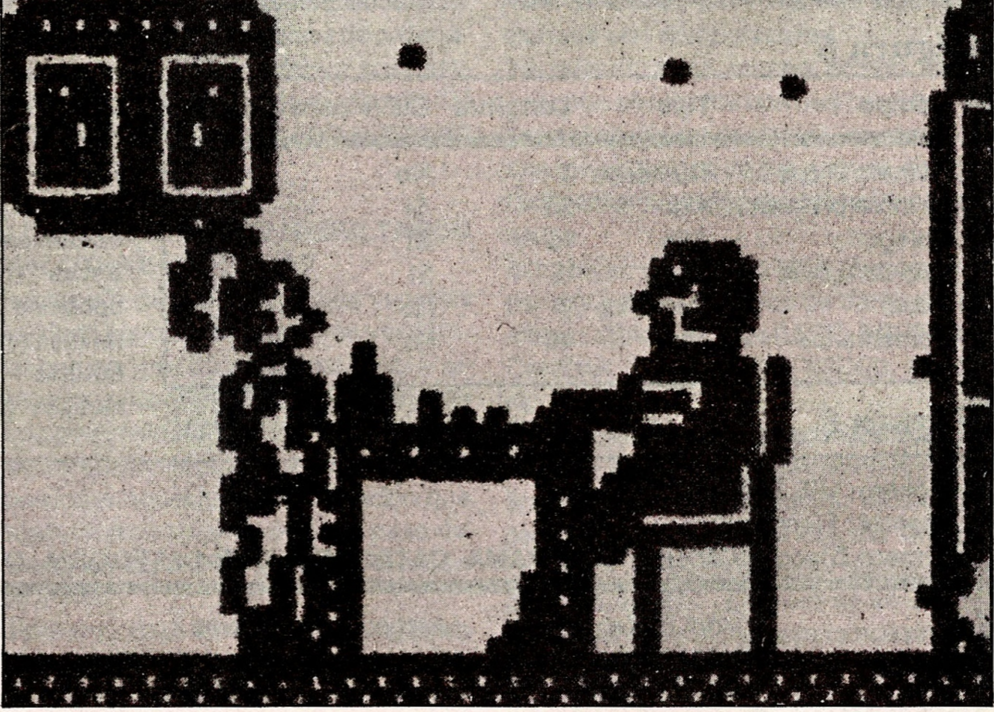
2.SPRITES: "rotar" gira la figura 90 grados a la izquierda o a la derecha. "Inverso" invierte la figura, es decir que lo que estaba en blanco pasará a ser negro y viceversa. "Espejo horizontal" o "vertical" modifica el sprite, creando la figura simétrica respecto a un eje horizontal o vertical. "Mover" mueve la figura una fila en la dirección indicada. "Cambiar", con tres posibilidades más: a) volver sprite a (1): para comenzar nuevamente con el sprite 1. b) ver sprites: permite chequear los sprites creados. c) sumar sprites: suma un sprite para poder trabajar con otro.

3.TAMAÑO DEL SPRITE: tenemos la posibilidad de ver nuestra creación en cinco tamaños diferentes.

4.DATOS: "binario", "decimal" o "hexadecimal". Los datos aparecen en pantalla en forma ordenada y en el sistema de numeración que especifiquemos. "Impresora" hace lo mismo que el anterior, pero tendremos una copia impresa. "Merge" permite intercalar los datos de los sprites en nuestros programas.

Para el uso del "merge" veamos un ejemplo:

- Terminemos de crear los sprites necesarios.
- Imgresemos en la opción "merge".
- Coloquemos los números de las líneas donde queremos ubicar los datos.
- Presionemos RETURN en cada una de las líneas para guardarlas en memoria.
- De esta manera estaremos en condiciones de mezclar el programa con los datos de los sprites utilizando MERGE "CAS:" o MERGE "nombre del programa", según necesitemos datasete o disquetera.



ESTRUCTURA DEL PROGRAMA

10-70: datos del autor.
80-220: inicializa pantalla y variables.
230-480: bucle principal-movimiento del cursor.
490-690: menú general.
700-760: dibuja figura-sprite pequeño.
770-970: imprime menú para cambiar el cuadro.
980-1090: velocidad del cursor.
1100-1260: menú de opción "SPRITES".
1270-1360: rotar.
1370-1450: inverso.
1460-1550: espejo horizontal.
1560-1650: espejo vertical.
1660-2020: menú de opción "MOVER".
2030-2180: menú para cambiar de sprite.
2190-2510: sumar un sprite; guardar sprite; ver sprite.
2520-2600: impresión de puntos.
2610-2750: tamaño del sprite.
2760-2950: ampliar-reducir sprite.
2960-3180: datos.
3190-3300: binario.
3310-3410: decimal.
3420-3520: hexadecimal.
3530-3730: imprimir.
3740-4080: merge.
4090-4310: cuadro borrar opciones cambiar color return letras.
4320-4700: información.
4880-5060: movimiento sprites.
5070-5260: presentación.
5270-5440: rotación sprites.
5450-5580: músicas.

5590-5740: definición de sprites.

LISTA DE VARIABLES

S$,P$: conjunto para establecer valores sobre las coordenadas.
CO: indica el color de los dos cuadros.
X%,Y%,X1%,Y1%: coordenadas para el desplazamiento del cursor.
D%: cambia el cursor para que borre o escriba.
MO%,Z,DI%: auxiliares utilizados en el movimiento de sprites.
OC%: indica el color de la mano.
IN%: verifica si ha entrado en la opción impresión.
FR%,TI%,LI%: indica el color de tinta, borde y frente de la palabra RETURN.
A%: movimiento de cursor mediante teclas.
C$,E$: variable auxiliar para utilizar el sistema binario.
V%: determina la velocidad del cursor.
A: direcciona lugares donde tiene que ir a partir del ingreso de datos mediante menúes.
A$: variable utilizada para el ingreso de datos mediante el teclado.
H: conjunto utilizado para la rotación de sprites.
M%,N%: variables utilizadas en lazos y auxiliares.
NS%: número de sprite.
G%: conjunto para cargar datos de sprites.

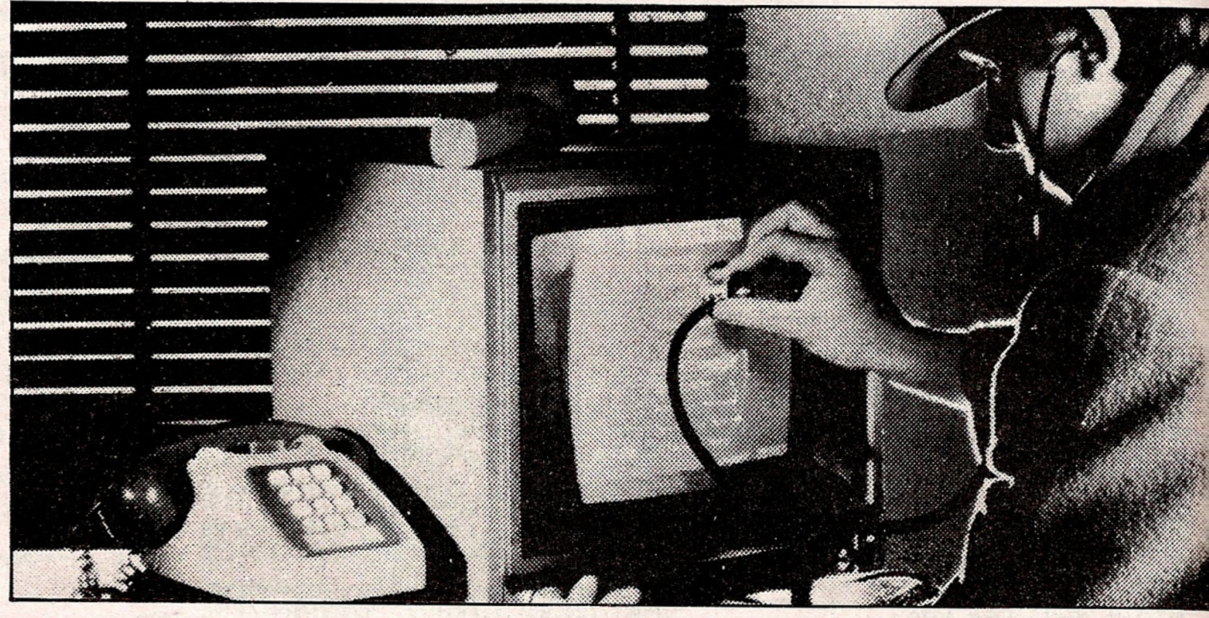
```
10 '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
20 '@                                   @
30 '@          S P R I T E S            @
40 '@                                   @
50 '@ DIEGO H.CONDE - SANTA ROSA - L.P @
60 '@       REVISTA << LOAD MSX >>      @
70 '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
80 SCREEN 2,2,0:COLOR 15,1,1:CLS:KEY OFF
:WIDTH 32
90 OPEN "GRP:" AS#1
100 '         INICIALIZACION-VARIABLES
110 DIM S$(16,16),P$(16,16),H(8),G%(480)
,L$(32)
120 GOSUB 5600
130 GOSUB 5070
140 CO=15:X%=9:Y%=9:D%=1:V%=1:MO%=0:DI%=
0:OC%=4:IN%=0:NS%=1
150 C$="1":GOSUB 970
160 GOSUB 5500
170 GOSUB 4100
190 GOSUB 4260
200 GOSUB 710
210 IF IN%=1 THEN IN%=0:GOTO 3720
220 FR%=13:TI%=1:LI%=11:GOSUB 4200
230 '
240 '             BUCLE PRINCIPAL
250 '
260 PLAY"T200M5000S206L9D05BGR1006EDT255
05L30"
270 PUT SPRITE 1,(X%*8-10,Y%*8+6),OC%,0
280 IF PLAY(0) THEN 280
290 ON STRIG GOSUB 460:STRIG(0) ON
300 A%=STICK(0)
310 IF INKEY$=CHR$(13) THEN 500
320 IF A%=0 THEN 300
330 X%=X%+SGN(A%-1)*SGN(5-A%):Y%=Y%+SGN(
A%-3)*SGN(7-A%)
340 IF X%<1 THEN X%=16
350 IF X%>16 THEN X%=1
360 IF Y%<3 THEN Y%=18
370 IF Y%>18 THEN Y%=3
380 IF D%=1 THEN E$="0" ELSE E$="1"
390 S$(Y%-2,X%)=E$
400 PSET (X%*8,Y%*8+4),D%:COLOR D%:PRINT
#1,"£":COLOR 15
410 PUT SPRITE 1,(X%*8-10,Y%*8+6),OC%,0
420 PSET(191+X%,6+Y%),D%:PLAY"M500S1V9A"
430 FOR N%=1 TO V%:NEXT N%
440 GOTO 290
450             CAMBIA CURSOR
460 IF D%=1 THEN D%=15:OC%=6:ELSE D%=1:O
C%=4
470 STRIG(0)ON:RETURN 380
480 RETURN
490 '
500 '             MENU GENERAL
510 '
520 PUT SPRITE 1,(0,209)
530 STRIG(0)OFF:GOSUB 4170
540 FR%=4:TI%=15:LI%=2:GOSUB 4200
550 PLAY"T200M5000S2V1205GB06DR1005G06C"
560 PSET(158,64):DRAW"C12E8R72F8D104G8L7
2H8U104"
570 PSET(159,64):DRAW"C12E5R71F8D104":PA
INT(168,57),12
580 COLOR 10:LINE(180,60)-(217,76),4,BF:
PSET(18?,64).1:PRINT#1,"MENU"
590 PSET(168,88),1:PRINT#1,"1-CUADRO":PS
ET(168,104),1:PRINT#1,"2-SPRITES":PSET(1
68,120),1:PRINT#1,"3-TAMA%O"
600 PSET(168,136),1:PRINT#1,"4-DATOS":PS
ET(168,152),1:PRINT#1,"5-INFORM."
610 COLOR 15
620 MO%=1:DI%=1
630 A$=INKEY$
640 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
240
650 IF A$<"1" OR A$>"5" THEN 630
660 PLAY"T255V120415M5000S3GB05DL4GR10DG
"
670 A=VAL(A$):FOR N=2 TO 12:LINE(163,67+
(A*16))-(240,85+(A*16)),N,B:LINE(164,68+
(A*16))-(239,84+(A*16)),N,B:NEXT N
680 PLAY"T120M5000S2L605DC04BR8L10AB05C0
4AG"
690 ON A GOTO 780,1110,2620,2970,4710
700 '
710 '          SPRITE PEQUE%O
720 '
730 LINE(192,9)-(207,24),CO,BF
740 LINE(188,5)-(211,28),6,B:LINE(189,6)
-(210,27),6,B
750 COLOR 15:LINE(160,35)-(255,43),1,BF:
PSET(160,35),1:PRINT#1,"SPRITE=";NS%
760 RETURN
770 '
780 '          CAMBIAR CUADRO
790 '
800 GOSUB 4170
810 FR%=6:TI%=15:LI%=10:GOSUB 4200
820 PSET(160,56):DRAW"C10R80D105H8G8H8G8
H8G8H8G8H8G8U105":PAINT(162,57),10
830 LINE(171,60)-(227,75),4,BF
840 PSET(176,64),4:PRINT#1,"CUADRO":COLO
R 1
850 PSET(168,88),10:PRINT#1,"1-BLANCO":P
SET(168,104),10:PRINT#1,"2-NEGRO":PSET(1
68,120),10:PRINT#1,"3-"
860 PSET(184,115),10:PRINT#1,"MANEJO":PS
ET(184,125),10:PRINT#1,"CURSOR":PSET(168
,136),10:PRINT#1,"4-MENU"
870 COLOR 15
880 A$=INKEY$
890 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
240
900 IF A$<"1" OR A$>"4" THEN 880
910 GOSUB 5460
920 A=VAL(A$):ON A GOTO 930,950,990,500
930 LINE(164,84)-(233,99),4,B:CO=15:C$="
1":D%=1:OC%=4:GOSUB 4140
940 GOSUB 710:GOSUB 970:GOSUB 4170:GOTO
240
950 LINE(164,100)-(233,113),4,B:CO=1:C$=
"0":D%=15:OC%=6:GOSUB 4140
960 GOSUB 710:GOSUB 970:GOSUB 4170:GOTO
240
970 FOR M%=1 TO 16:FOR N%=1 TO 16:S$(M%,
N%)=C$:P$(M%,N%)="":NEXT N%,M%:RETURN
980 '
990 '          VELOCIDAD CURSOR
1000 '
1010 GOSUB 4170
1020 PSET(158,58):DRAW"C2D30F10D45F10E10
F10E10F10E10U45E10U30L80":PAINT(159,59),
2
1030 LINE(158,56)-(238,57),6,BF:FOR N%=1
57 TO 238 STEP 6:LINE(N%,52)-(N%+3,55),6
,BF:NEXT N%
1040 COLOR1:PSET(165,70):PRINT#1,"VELOCI
DAD":PSET(190,86),2:PRINT#1,"DEL":PSET(1
78,102),2:PRINT#1,"CURSOR"
1050 PSET(181,120),2:PRINT#1,"(1-9)":COL
OR 15
1060 A$=INKEY$:IF A$="" THEN 1060
1070 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
1080 IF A$<"1" OR A$>"9" THEN PLAY"T1200
1V12L4C":GOTO 1060
1090 V%=VAL(A$)*5:GOSUB 4170:GOTO 240
1100 '
1110 '              SPRITES
1120 '
1130 FR%=1:TI%=15:LI%=10:GOSUB 4200
1140 GOSUB 4170
1150 PSET(162,61):FOR N=1 TO 5:DRAW"C13E
8F8":NEXT N
1160 DRAW"C13D120":FOR N=1 TO 5:DRAW"C13
G8H8":NEXT N:DRAW"C13U120":PAINT(165,61)
,13
1170 LINE(172,64)-(234,79),12,BF:PSET(17
6,69),12:PRINT#1,"SPRITES"
1180 COLOR 1
1190 PSET(168,88),13:PRINT#1,"1-ROTAR":P
SET(168,104),13:PRINT#1,"2-INVERSO":PSET
(168,120),13:PRINT#1,"3-ESPEJ.H"
1200 PSET(168,136),13:PRINT#1,"4-ESPEJ.V
":PSET(168,152),13:PRINT#1,"5-MOVER":PSE
T(168,168),13:PRINT#1,"6-CAMBIAR"
1210 COLOR 15
1220 A$=INKEY$
1230 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
1240 IF A$<"1" OR A$>"6" THEN 1220
1250 GOSUB 5460
1260 A=VAL(A$):LINE(164,68+(A*16))-(241,
84+(A*16)),1,B:LINE(164,68+(A*16))-(241,
84+(A*16)),1,B:ON A GOTO 1280,1380,1470,
1570,1670,2040
1270 '
1280 '                 ROTAR
1290 '
1300 GOSUB 4140:GOSUB 710
1310 FOR M%=1 TO 16:FOR N%=1 TO 16
1320 P$(M%,N%)=S$(17-N%,M%)
1330 SWAP S$(M%,N%),P$(M%,N%):GOSUB 2550
:SWAP P$(M%,N%),S$(M%,N%)
1340 NEXT N%,M%
1350 FOR M%=1 TO 16:FOR N%=1 TO 16:SWAP
S$(M%,N%),P$(M%,N%):NEXT N%,M%
1360 COLOR 15:GOSUB 4170:GOTO 240
1370 '
1380 '              INVERSO
1390 '
1400 FOR M%=1 TO 16:FOR N%=1 TO 16
1410 IF S$(M%,N%)="0" THEN S$(M%;N%)="1"
 ELSE S$(M%,N%)="0"
1420 GOSUB 2550
1430 NEXT N%,M%
1440 IF D%=1 THEN D%=15:CO=1:ELSE D%=1:C
O=15
1450 COLOR 15:GOSUB 4170:GOTO 240
```

```
1460 '
1470 '          ESPEJO HORIZONTAL
1480 '
1490 GOSUB 4140:GOSUB 710
1500 FOR M%=1 TO 16:FOR N%=1 TO 16
1510 P$(M%,N%)=S$(17-M%,N%)
1520 SWAP S$(M%,N%),P$(M%,N%):GOSUB 2550
:SWAP P$(M%,N%),S$(M%,N%)
1530 NEXT N%,M%
1540 FOR M%=1 TO 16:FOR N%=1 TO 16:SWAP
S$(M%,N%),P$(M%,N%):NEXT N%,M%
1550 COLOR 15:GOSUB 4170:GOTO 240
1560 '
1570 '          ESPEJO VERTICAL
1580 '
1590 GOSUB 4140:GOSUB 710
1600 FOR M%=1 TO 16:FOR N%=1 TO 16
1610 P$(M%,N%)=S$(M%,17-N%)
1620 SWAP S$(M%,N%),P$(M%,N%):GOSUB 2550
:SWAP P$(M%,N%),S$(M%,N%)
1630 NEXT N%,M%
1640 FOR M%=1 TO 16:FOR N%=1 TO 16:SWAP
S$(M%,N%),P$(M%,N%):NEXT N%,M%
1650 COLOR 15:GOSUB 4170:GOTO 240
1660 '
1670 '               MOVER
1680 '
1690 GOSUB 4170
1700 PSET(158,58):DRAW"C7R26E5H5R26G5F5R
31D15G7D76F10L91E10U76H7U15R3"
1710 COLOR 1:PAINT(159,59),7:LINE(175,64
)-(220,78),6,BF:PSET(179,67):PRINT#1,"MO
VER":COLOR 4
1720 PSET(168,88),7:PRINT#1,"1-ARRIBA":P
SET(168,104),7:PRINT#1,"2-ABAJO":PSET(16
8,120),7:PRINT#1,"3-DERECH"
1730 PSET(168,136),7:PRINT#1,"4-IZQUIE"
1740 A$=INKEY$
1750 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
1760 IF A$<"1" OR A$>"4" THEN 1740
1770 GOSUB 5460
1780 IF A$="2"OR A$="3"THEN MO%=0:DI%=-1
 ELSE MO%=1:DI%=1
1790 A=VAL(A$):LINE(165,68+(A*16))-(233,
84+(A*16)),1,B:ON A GOTO 1930,1930,1810,
1810
1800 '
1810 '          MOVER DER.IZQ
1820 '
1830 GOSUB 4140:GOSUB 710
1840 FOR M%=1 TO 16
1850 FOR N%=1 TO 16
1860 Z=N%+DI%
1870 IF MO%=0 AND Z=0 THEN Z=16 ELSE IF
Z=17 THEN Z=1
1880 P$(M%,N%)=S$(M%,Z)
1890 NEXT N%,M%
1900 FOR M%=1 TO 16:FOR N%=1 TO 16:SWAP
S$(M%,N%),P$(M%,N%):GOSUB 2550:NEXT N%,M
%
1910 COLOR 15:GOSUB 4170:GOTO 240
1920 '
1930 '          MOVER ABA.ARRI
1940 '
1950 GOSUB 4140:GOSUB 710
1960 FOR M%=1 TO 16
1970 FOR N%=1 TO 16
1980 Z=M%+DI%
1990 IF MO%=0 AND Z=0 THEN Z=16 ELSE IF
Z=17 THEN Z=1
2000 P$(M%,N%)=S$(Z,N%)
2010 NEXT N%,M%
2020 GOTO 1900
2030 '
2040 '          CAMBIAR DE SPRITE
2050 '
2060 GOSUB 4170
2070 PSET(152,61):DRAW"c4E4H4E4R15F5G5R5
8H5E5R15F5G5F3D102F2D7L99U7E2U102":PAINT
(153,62),4
2080 LINE(169,70)-(232,83),2,BF:PSET(176
,73),2:PRINT#1,"CAMBIAR"
2090 PSET(163,93),4:PRINT#1,"1-VOLVER":P
SET(170,105),4:PRINT#1,"SP. A (1)"
2100 PSET(163,120),4:PRINT#1,"2-VER":PSE
T(176,131),4:PRINT#1,"SPRITES"
2110 PSET(163,145),4:PRINT#1,"3-SUMAR":P
SET(176,156),4:PRINT#1,"SPRITE"
2120 A$=INKEY$:IF A$="" THEN 2120
2130 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
2140 IF A$<"1" OR A$>"3" THEN 2120
2150 PLAY"T120M5000S2L605DC04BR8L10AB05C
04AG"
2160 A%=VAL(A$):ON A% GOSUB 2180,2320,22
20
2170 GOSUB 4170:GOTO 220
2180 NS%=1:GOSUB 750:RETURN
2190 '
2200 '               SUMAR SP.
2210 '
2220 LINE(160,93)-(240,139),1,BF:PSET(16
3,100):PRINT#1,"GUARDANDO":PSET(175,115)
:PRINT#1,"SPRITE"
2230 '
2240 '               GUARDAR
2250 '
2260 B$="":E$="":GR%=1
2270 GOSUB 3320
2280 C$="1":GOSUB 970:CO=15:GOSUB 4140
2290 NS%=NS%+1:IF NS%>15 THEN NS%=15
2300 GR%=0:GOSUB 710:GOSUB 4170:GOTO 220
2310 '
2320 '               VER SPRITE
2330 '
2340 F$="":Q$=""
2350 LINE(160,145)-(240,166),1,BF:PSET(1
63,150),1:PRINT#1,"Nro.:"
2360 A$=INKEY$:IF A$="" THEN 2360
2370 IF A$<"1" OR A$>"9" THEN 2360
2380 BEEP:PSET(210,150),1:PRINT#1,A$
2390 F$=INKEY$:IF F$="" THEN 2390
2400 IF F$=CHR$(13) THEN F$="":GOTO 2430
2410 IF F$<"0" OR F$>"5" THEN 2390
2420 BEEP:PSET(218,150),1:PRINT#1,F$
2430 C$=A$+F$:IF VAL(C$)>15 THEN A$="":F
$="":LINE(210,150)-(226,158),1,BF:GOTO 2
360
2440 F$="":M%=(VAL(C$)-1)*32+1:FOR N%=M%
 TO M%+32:F$=F$+CHR$(G%(N%)):NEXT N%
2450 SPRITE$(15)=F$:CO=1:GOSUB 710:PUT S
PRITE 5,(192,8),15,15
2460 A$=INKEY$:IF A$="" THEN 2460
2470 PUT SPRITE 5,(0,209)
2480 FOR M%=1 TO 16:FOR N%=1 TO 16
2490 IF S$(M%,N%)="0" THEN CU%=1:C%=4 EL
SE CU%=15:C%=6
2500 GOSUB 2580:NEXT N%,M%
2510 RETURN
2520 '
2530 '               IMPRESION PUNTOS
2540 '
2550 PSET(N%*8,M%*8+20)
2560 IF S$(M%,N%)="0" THEN CU%=1:C%=4 EL
SE CU%=15:C%=6
2570 COLOR CU%:PRINT#1,"[":PUT SPRITE 1,
(N%*8-10,M%*8+20),C%,0
2580 X2%=191+N%:Y2%=8+M%
2590 PSET(X2%,Y2%),CU%
2600 RETURN
2610 '
2620 '               TAMA%O
2630 '
2640 GOSUB 4170
2650 FR%=13:TI%=15:LI%=11:GOSUB 4200
2660 PSET(158,58):DRAW"C2D30F10D53F10E10
F10E10F10E10U53E10U30L80":PAINT(159,59),
2
2670 LINE(158,56)-(238,57),6,BF:FOR N%=1
57 TO 238 STEP 6:LINE(N%,52)-(N%+3,55),6
,BF:NEXT N%
2680 COLOR1:PSET(168,70),2:PRINT#1," TAM
A%O  ":PSET(190,86),2:PRINT#1,"DEL":PSET
(178,102),2:PRINT#1,"SPRITE"
2690 PSET(180,120),2:PRINT#1,"(1-5)":COL
OR 15
2700 A$=INKEY$:IF A$="" THEN 2700
2710 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
2720 IF A$<"1" OR A$>"5" THEN PLAY"T1200
1V12L4C":GOTO 2700
2730 GOSUB 5460
2740 C%=VAL(A$)
2750 GOSUB 4170
2760 '
2770 '               AMPLIAR/REDUCIR
2780 '
2790 LINE(150,0)-(255,50),1,BF
2800 X2%=191-(C%*C%):Y2%=8
2810 FOR M%=1 TO 16:FOR N%=1 TO 16
2820 IF S$(M%,N%)="0" THEN CO=1 ELSE CO=
15
2830 LINE(X2%+N%-C%,Y2%+M%)-(X2%+N%,Y2%+
M%+C%-1),CO,BF
2840 X2%=X2%+C%-1
2850 NEXT N%:X2%=191-(C%*C%):Y2%=Y2%+C%-
1:NEXTM%
2860 X2%=191-(C%*C%):Y2%=8
2870 LINE (X2%-3-C%,5)-(X2%+(16*C%)+(5-C
%),Y2%+(16*C%)+4),15,B
2880 A$=INKEY$:IF A$<>CHR$(13) THEN 2880
2890 LINE(155,0)-(255,100),1,BF
2900 GOSUB 740
2910 FOR M%=1 TO 16:FOR N%=1 TO 16
2920 IF S$(M%,N%)="0" THEN CU%=1  ELSE C
U%=15
2930 GOSUB 2580
2940 NEXT N%,M%
2950 GOTO 240
2960 '
2970 '               DATOS
2980
2990 GOSUB 4170:B$="":E$="":GR%=0
3000 FR%=2:TI%=1:LI%=4:GOSUB 4200
3010 PSET(160,65):DRAW"C6R32H7R10E7F7R10
G7R32D116G10H10L16G6H6L16G10H10U116"
3020 PAINT(161,67),6
3030 LINE(181,69)-(224,83),12,BF
3040 PSET(184,72):PRINT#1,"DATOS"
3050 COLOR 1:PSET(167,97),6:PRINT#1,"1-B
INARIO":PSET(167,113),6:PRINT#1,"2-DECIM
AL":PSET(167,129),6:PRINT#1,"3-HEXADEC."
3060 PSET(167,145),6:PRINT#1,"4-IMPRIM."
:PSET(167,161),6:PRINT#1,"5-MERGE":COLOR
 15
3070 A$=INKEY$
3080 IF A$=CHR$(13) THEN GOSUB 4170:GOTO
 240
3090 IF A$<"1" OR A$>"5" THEN 3070
3100 GOSUB 5460
3110 A%=VAL(A$):LINE(161,77+(A%*16))-(24
3,93+(A%*16)),1,B
3120 IF A$="4" THEN IN%=1:GOTO 3540
3130 CO=1:GOSUB 4140
3140 PSET(71,28):DRAW"C4D127C1H63C15R127
"
3150 ON A% GOSUB 3200,3320,3430
3160 IF A%=4 THEN 3540
3170 IF A%=5 THEN 3750
3180 GOTO 3560
3190 '
3200 '               BINARIO
3210 '
3220 FOR M%=1 TO 16
3230 GOSUB 3280
3240 IF GR%=1 THEN LPRINT TAB(1);CHR$(14
);B$;CHR$(13):GOTO 3260
3250 PSET(9,21+M%*8),1:PRINT#1,B$
3260 B$="":NEXT M%
3270 RETURN
3280 '          CONVERSION BIN
3290 FOR T%=1 TO 16:B$=B$+S$(M%,T%):NEXT
 T%
3300 RETURN
3310 '
3320 '               DECIMAL
3330 '
3340 FOR M%=1 TO 16:GOSUB 3280:E$="&B"+L
EFT$(B$,8):Q$=STR$(VAL(E$))
3350 IF GR%=1 THEN G%(M%+NS%*16+(NS%-1)*
16-16)=VAL(Q$):GOTO 3370
3360 PSET(25,21+M%*8),1:PRINT#1,Q$
3370 E$="&B"+RIGHT$(B$,8):Q$=STR$(VAL(E$
))
3380 IF GR%=1 THEN G%(M%+NS%*16+(NS%-1)*
16)=VAL(Q$):GOTO 3400
3390 PSET(88,21+M%*8),1:PRINT#1,Q$
3400 B$="":E$="":NEXT M%
3410 RETURN
3420 '
3430 '               HEXADECIMAL
3440 '
3450 FOR M%=1 TO 16:GOSUB 3290:E$="&B"+L
EFT$(B$,8):Q$=RIGHT$("0"+HEX$(VAL(E$)),2
)
3460 IF GR%=1 THEN L$(M%)=Q$:GOTO 3480
3470 PSET(34,21+M%*8),1:PRINT#1,Q$
3480 E$="&B"+RIGHT$(B$,8):Q$=RIGHT$("0"+
HEX$(VAL(E$)),2)
3490 IF GR%=1 THEN L$(M%+16)=Q$:GOTO 351
0
3500 PSET(96,21+M%*8),1:PRINT#1,Q$
3510 B$="":Q$="":E$="":NEXT M%
3520 RETURN
3530 '
3540 '               IMPRIMIR
3550 '
3560 GOSUB 4170:B$="":E$="":GR%=0
3570 FR%=2:TI%=1:LI%=4:GOSUB 4200
3580 PSET(160,65):DRAW"C6R32H7R10E7F7R10
G7R32D92G10H10L16G6H6L16G10H10U92"
3590 PAINT(161,67),6
3600 LINE(173,69)-(243,83),12,BF
3610 PSET(176,72),12:PRINT#1,"IMPRIMIR"
3620 COLOR 1:PSET(167,97),6:PRINT#1,"1-B
INARIO":PSET(167,113),6:PRINT#1,"2-DECIM
AL":PSET(167,129),6:PRINT#1,"3-HEXADEC."
:COLOR 15
3630 A$=INKEY$
3640 IF A$=CHR$(13) THEN 3690
3650 IF A$<"1" OR A$>"3" THEN 3630
3660 GOSUB 5460
3670 A%=VAL(A$):LINE(161,77+(A%*16))-(24
3,93+(A%*16)),1,B
3680 GOTO 4320
3690 GOSUB 4170
3700 IF IN%=1 THEN IN%=0:GOTO 240
3710 GOSUB 4140
3720 FOR M%=1 TO 16:FOR N%=1 TO 16:GOSUB
 2550:NEXT N%,M%
3730 GOTO 220
3740 '
```

```
3750 '              MERGE
3760 '
3770 CO=1:GOSUB 4140
3780 PSET(10,30),1:PRINT#1,"MERGE:":PSET
(10,45),1:PRINT#1,"PERMITE INTER-":PSET(
10,53),1:PRINT#1,"CALAR LOS DATOS"
3790 PSET(10,61),1:PRINT#1,"DE LOS SPRIT
ES":PSET(10,69),1:PRINT#1,"EN SU PROGRAM
A."
3800 PSET(10,83),1:PRINT#1,"<SU PROGRAMA
 DE_":PSET(10,91),1:PRINT#1,"BE ESTAR GR
ABADO":PSET(10,99),1:PRINT#1,"EN ASCII."
3810 PSET(10,113),1:PRINT#1,"EL PROGRAMA
":PSET(10,121),1:PRINT#1," <<-SPRITES >>
":PSET(10,129),1:PRINT#1,"SE AUTODESTRUI
RA"
3820 PSET(18,143),1:PRINT#1,"CONTINUA (S
/N)"
3830 A$=INKEY$:IF A$="" THEN 3830
3840 IF A$="S" OR A$="s" THEN 3880
3850 CO=15:GOSUB 4140
3860 FOR M%=1 TO 16:FOR N%=1 TO 16:GOSUB
 2550:NEXT N%,M%
3870 GOSUB 4170:GOTO 220
3880 LINE(18,143)-(138,151),1,BF:PSET(18
,143),1:PRINT#1,"UN MOMENTO!!!":B$="":E$
="":GR%=1:GOSUB 3320
3890 SCREEN 1:WIDTH 32:CLS:B$=""
3900 CLS:LOCATE 0,0:PRINT"Nro. DE COMIEN
ZO DE LAS LINEAS":LOCATE 0,1:PRINT"DE LO
S DATOS (MAYOR DE 9999)"
3910 A%=0:LOCATE 10,3:LINE INPUT A$
3920 FOR N%=1 TO NS%:LOCATE 0,N%*5
3930 IF N%>4 THEN 4030
3940 PRINT VAL(A$)+N%*10-10;"DATA ";
3950 FOR M%=1 TO 32:A%=A%+1:B$=STR$(G%(A
%)):PRINT MID$(B$,2,3);",";:NEXT M%
3960 A=POS(X):B=CSRLIN:LOCATE A-1,B:PRIN
T" ":NEXT N%
3970 PRINT"GOTO 3980":GOTO 4080
3980 CLS:LOCATE 0,0:PRINT"DELETE 10-3980
":PRINT:PRINT"DELETE 4090-5740":PRINT:PR
INT"GOTO 3990":END
3990 CLS:LOCATE 0,0:PRINT"PARA MEZCLAR S
U PROGRAMA TECLEE:"
4000 PRINT:PRINT"MERGE";CHR$(34);"CAS:";
CHR$(34);" .....PARA DATASSETTE"
4010 PRINT"MERGE";CHR$(34);"NOMBRE DEL P
ROG.";CHR$(34);" ...PARA DISKETERA"
4020 DELETE 3990-4080:END
4030 IF N%=5 THEN 4070
4040 IF N%=9 THEN 4070
4050 IF N%=13 THEN 4070
4060 GOTO 3940
4070 LOCATE 0,23:PRINT"CLS:LOCATE 0,24:G
OTO 3940"
4080 LOCATE 0,24:PRINT"*PRESIONE RETURN
EN CADA LINEA*":END
4090 '
4100 '             CUADRO
4110 '
4120 LINE(0,20)-(144,163),4,B:LINE(1,21)
-(143,162),4,B
4130 LINE(2,22)-(142,161),4,B:LINE(4,24)
-(140,159),4,B
4140 LINE(8,28)-(135,155),CO,BF
4150 LINE(7,27)-(136,156),4,B
4160 RETURN
4170 '         BORRA OPCIONES
4180 LINE(145,47)-(255,192),1,BF
4190 RETURN
4200 '       CAMBIA COLOR RETURN
4210 LINE(0,174)-(164,186),1,BF
4220 LINE(0,174)-(62,186),FR%,BF
4230 COLOR TI%:PSET(8,177):PRINT#1,"RETU
RN":COLOR 15:PSET(68,177),1:PRINT#1,"par
a salir"
4240 LINE(1,173)-(63,187),LI%,B:LINE(0,1
72)-(64,188),LI%,B
4250 RETURN
4260 '         LETRAS AMPLIADAS
4270 PRESET(0,180):PRINT#1,"SPRITES >"
4280 FOR N%=0 TO 80:FOR M%=180 TO 192:T=
T+1
4290 IF POINT(N%,M%)=15 THEN LINE(N%*2,T
*2)-(N%*2+6,T*2+2),15,B
4300 NEXT M%:T=0:NEXT N%
4310 RETURN
4320 '
4330 '            IMPRESION
4340 '
4350 LPRINT:LPRINT STRING$(79,"-")
4360 LPRINT CHR$(14);TAB(10)"DATOS DE LO
S SPRITES";CHR$(13)
4370 LPRINT:LPRINT
4380 IF A%=1 THEN A$="BINARIO"
4390 IF A%=2 THEN A$="DECIMAL"
4400 IF A%=3 THEN A$="HEXADECIMAL"
4410 LPRINT:LPRINT"SISTEMA NUMERICO: ";A
$
4420 LPRINT"Sprite Nro.: ";NS%:LPRINT
4430 GOSUB 4170
4440 LINE(165,68)-(240,118),4,BF
4450 PSET(172,72),1:PRINT#1,"ESPERE":PSE
T(190,88),1:PRINT#1,"UN":PSET(172,104),1
:PRINT#1,"MOMENTO"
4460 GR%=1:IF A%=1 THEN GOSUB 3220:GOTO
4600
4470 A$="--------------------":LPRINT CHR
$(14);A$
4480 ON A% GOSUB ,3340,3450
4490 IF A%=2 THEN 4510
4500 IF A%=3 THEN 4610
4510 FOR M%=1 TO 16
4520 LPRINT CHR$(14);"!";
4530 LPRINT TAB(3);USING"###";G%(M%+NS%*
16+(NS%-1)*16-16);
4540 LPRINT CHR$(14);TAB(9);"!";
4550 LPRINT TAB(13);USING"###";G%(M%+NS%
*16+(NS%-1)*16);
4560 LPRINT CHR$(14);TAB(18);"!"
4570 IF M%=8 THEN LPRINT CHR$(14);A$
4580 NEXT M%
4590 LPRINT CHR$(14);A$
4600 GR%=0:GOTO 3690
4610 FOR M%=1 TO 16
4620 LPRINT CHR$(14);"!";
4630 LPRINT TAB(4);L$(M%);
4640 LPRINT CHR$(14);TAB(8);"!";
4650 LPRINT TAB(12);L$(M%+16);
4660 LPRINT CHR$(14);TAB(18);"!"
4670 IF M%=8 THEN LPRINT CHR$(14);A$
4680 NEXT M%
4690 LPRINT CHR$(14);A$
4700 GOTO 4600
4710 '
4720 '            INFORMACION
4730 '
4740 CLS
4750 LINE(6,6)-(30,186),11,BF:LINE(0,0)-
(36,192),6,B:LINE(1,1)-(35,191),6,B
4760 PSET(40,0),1:PRINT#1,"INFORMACION:"
4770 PSET(40,24),1:PRINT#1,"1-CUADRO  NE
GRO."
4780 PSET(120,16),1:PRINT#1,"BLANCO.":PS
ET(120,32),1:PRINT#1,"MANEJO CURSOR."
4790 PSET(116,10),15:DRAW"G3D10G3F3D10F3
"
4800 PSET(120,48),1:PRINT#1,"ROTAR.":PSE
T(120,56),1:PRINT#1,"INVERSO."
4810 PSET(40,64),1:PRINT#1,"2-SPRITES ES
P.HORIZONTAL.":PSET(120,72),1:PRINT#1,"E
SPEJ.VERICAL"
4820 PSET(117,46),15:DRAW"G3D15G3F3D15F3
":PSET(120,80),1:PRINT#1,"CAMBIAR=0/+/VE
R"
4830 PSET(120,88),1:PRINT#1,"MOVER.":PSE
T(74,112),1:PRINT#1,"ARRIBA - ABAJO":PSE
T(74,120),1:PRINT#1,"DERECHA-IZQUIERDA"
4840 PSET(71,108),15:DRAW"E3R61E3F3R61F3
"
4850 PSET(40,136),1:PRINT#1,"3-TAMA%O DE
L SPRITE."
4860 PSET(120,152),1:PRINT#1,"BINARIO.":
PSET(120,160),1:PRINT#1,"HEXADECIMAL.":P
SET(40,168),1:PRINT#1,"4-DATOS  DECIMAL
.":PSET(120,176),1:PRINT#1,"IMPRIMIR."
4870 PSET(120,184),1:PRINT#1,"MERGE.":PS
ET(116,148),15:DRAW"G3D15G3F3D15F3"
4880 '
4890 '           MOV.SPRITES
4900 '
4910 M%=0:N%=5:IN%=1
4920 IF M%=0 THEN N%=N%+1:IF N%=23 THEN
M%=1:GOTO 4940:ELSE 4950
4930 GOTO 4950
4940 A$=INKEY$:IF A$="" THEN 4950 ELSE 5
060
4950 IF M%=1 THEN N%=N%-1:IF N%=5 THEN M
%=0
4960 PUT SPRITE 1,(10,N%),6 ,1
4970 PUT SPRITE 2,(10,N%*2-2),4 ,2
4980 PUT SPRITE 3,(10,N%*3-3),6 ,3
4990 PUT SPRITE 4,(10,N%*4-4),4 ,4
5000 PUT SPRITE 5,(10,N%*5-5),6 ,5
5010 PUT SPRITE 6,(10,N%*6-6),4 ,6
5020 PUT SPRITE 7,(10,N%*7-7),6 ,7
5030 PUT SPRITE 8,(10,N%*7-7),4 ,1
5040 FOR M=1 TO 10:NEXT M
5050 GOTO 4920
5060 FOR N=1 TO 8:PUT SPRITE N,(0,209):N
EXT N:CLS:GOTO 170
5070 '
5080 '          PRESENTACION
5090 '
5100  LINE(128,64)-(128,192),3
5110  FORI=1TO6:LINE(128-I*8,64)-(128-I*
20,191),3:LINE(128+I*8,64)-(128+I*20,191
),3:NEXT
5120 RESTORE 5130:FORI=1TO7:READA:LINE(1
78+I*8,64)-(255,A),3:LINE(82-I*8,64)-(0,
A),3:NEXT
5130  DATA 170,142,121,104,90,78,70
5140  FORI=2TO5STEP.3:Y=EXP(I)+57:LINE(0
,Y)-(255,Y),3:NEXT
5150  FORI=0TO60:X=INT(RND(1)*256):Y=INT
(RND(1)*63):C=INT(RND(1)*15+1):PSET(X,Y)
,C:NEXT
5160 CIRCLE(0,0),50,10:PAINT(1,1),10
5170 SOUND 7,&B11111110:SOUND 8,15:B=50:
C=200:FOR N=7 TO 1 STEP -1:C=C-20:B=B-4
5180 FOR M=0 TO C:PUT SPRITE N,(M,B),15,
N:SOUND 0,M:NEXT M:NEXT N
5190 SOUND 8,0
5200 GOSUB 5540
5210 CIRCLE(200,137),21,4:PAINT(200,137)
,4
5220 PSET(200,154):DRAW"u1h1u2h1u2h11h1
u1h1u1e2u1h21h2u1h1u2e2u1h212g1"
5230 DRAW"u1e2r2e1r2"
5240 PSET(200,151):DRAW"r1u3e2u2h2u11h2
11h2e2r1f1d1u3e1u1e1u2e1u1h2g21h1e2h112
"
5250 PAINT(201,147),15:PAINT(200,130),15
5260 A$="POR DIEGO CONDE":PSET(20,180),1
:PRINT#1,A$:PSET(21,180),1:PRINT#1,A$
5270 '
5280 '             ROTACION
5290 '
5300 FOR M=1 TO 8:H(M)=M-1:NEXT M
5310 FOR P=0 TO 7:FOR N=1 TO 32
5320 PUT SPRITE H(1),(161,96+N),15,H(1)
5330 PUT SPRITE H(2),(161,128+N),H(2)+2,
H(2)
5340 PUT SPRITE H(3),(161+N,160),H(3)+2,
H(3)
5350 PUT SPRITE H(4),(193+N,160),H(4)+2,
H(4)
5360 PUT SPRITE H(5),(225,160-N),H(5)+2,
H(5)
5370 PUT SPRITE H(6),(225,128-N),H(6)+2,
H(6)
5380 PUT SPRITE H(7),(225-N,96),H(7)+2,H
(7)
5390 PUT SPRITE H(8),(193-N,96),H(8)+2,H
(8)
5400 A$=INKEY$:IF A$<>"" THEN FOR N=0 TO
 15:PUT SPRITE N,(0,209):NEXT N:CLS:BEEP
:RETURN
5410 NEXT N
5420 PROV=H(8)
5430 FOR T=8 TO 2 STEP-1:H(T)=H(T-1):NEX
T T
5440 H(1)=PROV:GOTO 5310
5450 '
5460 '              MUSICAS
5470 '
5480 PLAY"T220M5000S2L606D05BL4GGR20L6GA
B06CL3DDD05B"
5490 RETURN
5500 A$="T25504V10M5000S113GB05DL3GL5GGG
L3R8DL5DDD04L3R8B05D04BGR2"
5510 B$="T25503V12M5000S1L3GB04DL3GL5GGG
L3R8DL5DDD03L3R8B04d03BGR2"
5520 PLAY A$,B$
5530 RETURN
5540 A$="t180v10m7000s2o415gggg14br11bo5
13dr1014dedc13do4br415gggg14br11bo513dr1
015dedc13dr5"
5550 B$="15o4ggggo512f#13dr5"
5560 C$="15o4ggggo5dcdo413o5eo4r514ggggg
11g"
5570  PLAY A$+B$:PLAY B$+C$
5580 RETURN
5590 '
5600 '        DEFINICION DE SPRITES
5610 '
5620 RESTORE 5680:FOR M%=0 TO 6:FOR N%=1
 TO 32
5630 READ A$:B$=B$+CHR$(VAL("&H"+A$)):NE
XT N%
5640 SPRITE$(M%)=B$
5650 A$="":B$="":NEXT M%
5660 SPRITE$(7)=SPRITE$(1)
5670 RETURN
5680 DATA 00,00,00,06,02,0E,0F,1F,1F,1F,
0F,0F,03,00,00,00,06,0A,1E,3C,58,F0,F0,E
C,F6,FB,FE,FC,F8,E0,00,00
5690 DATA 07,0C,18,18,10,0F,00,00,08,10,
18,1F,07,00,00,00,E0,30,10,20,00,E0,30,1
0,10,30,70,F0,C0,00,00,00
5700 DATA 00,0F,1C,18,18,18,18,18,1F,1E,
18,18,30,30,60,00,00,E0,70,10,10,10,30,6
0,C0,00,00,00,00,00,00,00
5710 DATA 00,0F,1C,18,18,18,18,1C,1F,19,
18,18,10,30,00,00,00,C0,60,20,20,20,20,C
0,80,C0,40,60,20,38,00,00
5720 DATA 00,01,03,01,01,01,01,01,01,01,
01,03,07,03,00,00,00,80,C0,80,80,00,00,0
0,00,00,00,80,C0,80,00,00
5730 DATA 00,0F,3F,13,01,01,01,01,01,01,
01,03,07,03,00,00,00,E0,F0,98,10,00,00,0
0,00,00,00,80,C0,80,00,00
5740 DATA 00,07,0E,0C,0C,0C,0F,0F,0C,0C,
0C,08,07,03,00,00,00,E0,30,00,00,80,C0,0
0,00,00,60,30,E0,80,00,00
```

EL JUEGO DE LOS DETECTIVES

Autor: Esteban García
Tipo: Juego
Participa del 4º concurso de programas



El juego de los detectives, que da origen a este programa escrito en BASIC, consiste en descubrir tres cartas desconocidas (habitación, arma, asesino), cada una de ellas extraída al azar de series de nueve. Las 24 cartas restantes, una vez barajadas, son repartidas entre tres jugadores, que en este caso son el operador y dos jugadores movidos por la máquina. Los jugadores tienen, a su turno, que desplazarse por un tablero y, de acuerdo a la habitación en que se encuentran, formular una triple pregunta: "sospecha". El siguiente jugador debe mostrar una de esas tres cartas, si la tiene, pero sólo al que pregunta, en caso contrario la obligación es del otro jugador. De esta manera se van descartando los posibles culpables hasta llegar a poder afirmar con certeza cuáles son las cartas ocultas, teniendo en cuenta siempre la obligación de estar en la habitación correspondiente. En caso de no acertar al "afirmar", se pierde el juego.

Para agilizar el juego y evitar errores en el ingreso de datos, el operador participa mediante menúes controlados por las teclas del cursor y la barra espaciadora o el joystick.

La rutina principal del programa consiste en un bucle cerrado, dividido en bloques que pueden alternarse según qué jugador responde, pero volviendo a la secuencia del juego mediante el flag "H" que devuelve la ejecución al turno que corresponde.

Cada uno de los jugadores controlados por la máquina (2)-(Holmes) y (3)-(Poirot), poseen una subrutina que comprueba y anota las respuestas. Un contador para cada tipo de serie suma los aciertos, poniéndose 8 en caso de no obtener respuesta de los otros jugadores y no poseerla él tampoco. Cuando la suma de los contadores de un jugador es igual a 24 significa que ha descubierto a los culpables y está en condiciones de afirmar. Para ello necesita estar en la habitación en que se cometió el crimen. Se dirige a ella mediante una subrutina que calcula la ruta más corta y, una vez allí, va a una rutina de "final".

Los jugadores (2) y (3) se desplazan por el tablero durante el juego por medio de rutas almacenadas en "DATAS" y elegidas al azar.

El jugador que opera la máquina, al optar por "afirmar", se dirige a una subrutina que comprueba la veracidad, y depende de ello la rutina final.

ESTRUCTURA DEL PROGRAMA

140-170: Desviaciones a subrutinas de presentación, instrucciones, carga de matrices y dibujo del tablero.
230-780: JUEGA (1)-(operador)
240-260: Define flag, contadores, etcétera
270-380: Opción "sospecha"-"afirma"
390-540: Selecciona habitación
550-780: Elige arma y sospechoso
790-990: CONTESTA (2) (Holmes)
1000-1250: JUEGA(2) (Holmes)
1260-1450: CONTESTA (3) (Poirot)
1460-1720: JUEGA (3) (Poirot)
1730-2120: CONTESTA (3) (operador)
1730-1910: Menú de respuesta
1939-2100: Comprueba respuesta
2110-2120: Cierre del bucle principal
2130-2600: Anota respuestas jugadores (2) y (3)
2610-2980: Elige ruta más corta a habitación culpable
2990-3040: Comprueba acusación jugador (1)
3050-3340: Final-Afirmación equivocada
3370-3650: Final-Afirmación cierta
3660-4000: Instrucciones y selección cursor o joystick
4010-4730: Carga de matrices
4740-5270: Dibuja tablero
5280-5670: Carátula presentación
5680-6250: DATAS

LISTA DE VARIABLES NUMERICAS

Q= contador de jugadas.
R= contador. Fija secuencia de ruta.
H= flag. Controla qué jugador efectuó la pregunta.
SD= flag. Indica si el operador afirma o

sospecha.
W= stick.
JO= indica si juega con joystick o cursor.
C= número de cuarto culpable.
ZH= es igual a la suma de los valores de la matriz Z.
KP= es igual a la suma de los valores de la matriz K.
N2= guarda número de habitación en que se encuentra (2).
N3= guarda número de habitación en que se encuentra (3).
R2= indica el número de ruta del jugador (2).
R3= indica el número de ruta del jugador (3).
N= carga indistintamente el valor de N2 o N3 para uso de la rutina ruta más corta a habitación culpable.
YM= coordenada Y del sprite "mano".
X1, Y1= coordenadas X e Y del sprite del jugador 1.
X2, Y2= coordenadas X e Y del sprite del jugador 2.
X3, Y3= coordenadas X e Y del sprite del jugador 3.
S= número RND.

LISTA DE VARIABLES ALFANUMERICAS

A1$, A2$,A3$= música.
A$= Draw.
B$= mensaje en pantalla presentación.

LISTA DE MATRICES NUMERICAS

z(3)= cuenta los aciertos de cada una de las series usadas por jugador (2).
K(3)= cuenta los aciertos de cada una de las series usadas por jugador (3).
RU(9,10)= en esta matriz se encuentran 9 rutas posibles de 10 elementos cada una.
X(9)= guarda las coordenadas X de las habitaciones.
Y(9)= guarda las coordenadas Y de las habitaciones.

MATRICES ALFANUMERICAS

SO$(3)= carga las sospechas.
L$(27)= carga las 27 cartas. Después de descontar los 3 culpables, reparte mezcladas 8 cartas a cada jugador.
F$(3,9)= carga las 27 clasificadas en series: habitación, arma, sospechoso.
**G2$(3,9)=** similar a F$. De ella va borrando las cartas que le muestran al jugador (2).
**G3$(3,9)=** similar a F$. De ella va borrando las cartas que le muestran al jugador (3).
C$(3)= almacena los culpables.
J1$(8)= almacena cartas jugador (1).
J2$(8)= almacena cartas jugador (2).
J3$(8)= almacena cartas jugador (3).

```
10 ' ****************************
20 ' *                          *
30 ' * EL JUEGO DE LOS DETECTIVES *
40 ' *                          *
50 ' *          POR             *
60 ' *                          *
70 ' *    ESTEBAN  H. GARCIA    *
80 ' *                          *
90 ' *         1988             *
100 '*                          *
110 '****************************
120 '
130 '
140 GOSUB5340:'PANTALLA PRESENT.
150 GOSUB3660:'INSTRUCCIONES
160 GOSUB4010:'CARGA MATRICES
170 GOSUB4740:'DIB. TABLERO
180 '   *************************
190 '   *                       *
200 '   * RUT. PRINCIPAL DE JUEGO *
210 '   *                       *
220 '   *************************
230 ' JUEGA(1)
240 Q=Q+1:H=1
250 R=R+1:IFR=11THENR=1
260 YM=20:COLOR 1
270 LINE(150,0)-(250,135),12,BF
280 LINE(150,0)-(250,135),15,B
290 PSET(170,90):PRINT#1,"JUGADA":PSET(1
80,100):PRINT#1,"Nx";Q
300 PSET(170,60):PRINT#1,"AFIRMAR "
310 PSET(170,20):PRINT#1,"SOSPECHAR":COL
OR 15
320 IFSTICK(JO)=1THENYM=YM-40:SD=0
330 IFYM=<20THENYM=20
340 IFSTICK(JO)=5THENYM=YM+40:SD=1
350 IFYM=>60THENYM=60
360 PUTSPRITE0,(151,YM-3),1,0
370 IFSTRIG(JO)=0THEN320
380 PUTSPRITE0,(0,209),15,0
390 LINE(152,10)-(247,80),12,BF
400 PSET(160,20):PRINT#1,"JUEGAS TU:"
410 PSET(160,40):PRINT#1,"DIRIGETE A"
420 PSET(160,60):PRINT#1,"UN CUARTO"
430 W=STICK(JO):IFW=0THEN440ELSE450
440 IFSTRIG(JO)=0THEN430ELSE540
450 IFY1=20THEN470
460 IFW=1THENY1=Y1-45:N1=N1-3
470 IFX1=120THEN490
480 IFW=3THENX1=X1+45:N1=N1+1
490 IFY1=110THEN510
500 IFW=5THENY1=Y1+45:N1=N1+3
510 IFX1=30THEN530
520 IFW=7THENX1=X1-45:N1=N1-1
530 PUT SPRITE1,(X1-15,Y1),1,1
540 SO$(1)=F$(1,N1)
550 '
560 'ELIGE ARMA Y SOSPECHOSO
570 '
580 A$=" ARMA"
590 FORA=2TO3
600 LINE(150,0)-(250,135),12,BF
610 LINE(150,0)-(250,135),15,B
620 PSET(160,10):PRINT#1,"JUEGAS TU:"
630 PSET(160,20):PRINT#1," ELIGE UN"
640 PSET(160,30):PRINT#1,A$
650 FORU=1TO9
660 PSET(166,U*10+30):PRINT#1,F$(A,U)
670 NEXTU
680 IFSTICK(JO)=1THENYM=YM-10:FORT=1TO50
:NEXT T
690 IFYM=<40THENYM=40
700 IFSTICK(JO)=5THENYM=YM+10:FORT=1TO50
:NEXT T
710 IF YM=>120 THENYM=120
720 PUT SPRITE0,(150,YM-3),1,0
730 IFSTRIG(JO)=0THEN680
740 SO$(A)=F$(A,(YM-30)/10)
750 A$="SOSPECHOSO"
760 NEXTA
770 PUTSPRITE0,(0,209),1,0
780 IFSD=1THENGOTO 2980
790 '
800 'CONTESTA(2)
810 '
820 IFH=2THENGOTO2320
830 LINE(150,0)-(250,135),12,BF
840 LINE(150,0)-(250,135),15,B
850 PSET(163,20):PRINT#1,"SHERLOCK"
860 PSET(163,30):PRINT#1," HOLMES"
870 PSET(163,40):PRINT#1,"CONTESTA"
880 FORU=1TO3:FORA=1TO8
890 IFSO$(U)=J2$(A)THENGOTO940
900 NEXTA:NEXTU
910 PSET(163,60):PRINT#1,"NO TENGO"
920 PSET(163,70):PRINT#1,"NINGUNO"
930 GOSUB 1250
940 PSET(163,55):PRINT#1,"TENGO"
950 IFH<>1THEN970
960 PSET(163,70):PRINT#1,J2$(A)
970 IFH=3THENGOSUB 2470:GOTO 240
980 IFSTRIG(JO)=0THEN980
990 '
1000 'JUEGA(2)
1010 '
1020 H=2:ZH=Z(1)+Z(2)+Z(3)
1030 IFZH=24THENN=N2:GOTO 2620
1040 N2=RU(R2,R)
1050 SO$(1)=F$(1,N2)
1060 FORA=2TO3
1070 T=INT(RND(-TIME)*9)+1
1080 IFG2$(A,T)="I"THEN1070
1090 SO$(A)=F$(A,T)
1100 NEXTA
1110 X2=X(N2):Y2=Y(N2)
1120 PUT SPRITE2,(X2,Y2),1,2
1130 LINE(150,0)-(250,135),12,BF
1140 LINE(150,0)-(250,135),15,B
1150 PSET(163,20):PRINT#1,"SHERLOCK"
1160 PSET(163,30):PRINT#1," HOLMES"
1170 PSET(163,40):PRINT#1," JUEGA:"
1180 LINE(160,48)-(230,59),11,BF:COLOR 1
1190 IFZH=24ANDN2=CTHENPSET(163,50):PRIN
T#1,"AFIRMO":GOTO 1210
1200 PSET(163,50):PRINT#1,"SOSPECHO"
1205 COLOR 15
1210 PSET(163,70):PRINT#1,SO$(1)
1220 PSET(163,80):PRINT#1,SO$(2)
1230 PSET(163,90):PRINT#1,SO$(3)
1240 IFZH=24ANDN2=CTHENFORT=1TO3000:NEXT
T:GOTO 3350
1250 IFSTRIG(JO)=0THEN1250
1260 'CONTESTA(3)
1270 '
1280 IFH=3THENGOTO 2550
1290 LINE(150,0)-(250,135),12,BF
1300 LINE(150,0)-(250,135),15,B
1310 PSET(163,20):PRINT#1,"HERCULES"
1320 PSET(163,30):PRINT#1," POIROT"
1330 PSET(163,40):PRINT#1,"CONTESTA"
1340 FORU=1TO3:FORA=1TO8
1350 IFSO$(U)=J3$(A)THENGOTO1400
1360 NEXTA:NEXTU
1370 PSET(163,60):PRINT#1,"NO TENGO"
1380 PSET(163,70):PRINT#1,"NINGUNO"
1390 GOTO  1720
1400 PSET(163,55):PRINT#1," TENGO"
1410 IFH<>1THEN1430
1420 PSET(163,70):PRINT#1,J3$(A)
1430 IFH=2THENGOSUB2130
1440 IFH=1THENGOTO 980
1450 IFSTRIG(JO)=0THEN1450
1460 '
1470 'JUEGA(3)
1480 '
1490 H=3:KP=K(1)+K(2)+K(3)
1500 IFKP=24THENN=N3:GOTO2620
1510 N3=RU(R3,R)
1520 SO$(1)=F$(1,N3)
1530 FORA=2TO3
1540 T=INT(RND(-TIME)*9)+1
1550 IFG3$(A,T)="I"THEN1540
1560 SO$(A)=F$(A,T)
1570 NEXTA
1580 X3=X(N3):Y3=Y(N3)
1590 PUT SPRITE3,(X3,Y3),1,3
1600 LINE(150,0)-(250,135),12,BF
1610 LINE(150,0)-(250,135),15,B
1620 PSET(163,20):PRINT#1,"HERCULES"
1630 PSET(163,30):PRINT#1," POIROT"
1640 PSET(163,40):PRINT#1," JUEGA:"
1650 LINE(160,48)-(230,59),11,BF:COLOR 1
1660 IFKP=24ANDN3=CTHENPSET(163,50):PRIN
T#1,"AFIRMO":GOTO 1680
1670 PSET(163,50):PRINT#1,"SOSPECHO"
1675 COLOR 15
1680 PSET(163,70):PRINT#1,SO$(1)
1690 PSET(163,80):PRINT#1,SO$(2)
1700 PSET(163,90):PRINT#1,SO$(3)
1710 IFKP=24ANDN3=CTHENFORT=1TO3000:NEXT
T:GOTO 3350
1720 IFSTRIG(JO)=0THEN1720
```

```
1730 '
1740 'CONTESTA(1)
1750 '
1760 IFH=1THEN980
1770 LINE(150,0)-(250,135),12,BF
1780 LINE(150,0)-(250,135),15,B
1790 PSET(153,20):PRINT#1,"CUAL TIENES?"
1800 PSET(166,40):PRINT#1,"NO TENGO"
1810 PSET(166,60):PRINT#1,SO$(1)
1820 PSET(166,80):PRINT#1,SO$(2)
1830 PSET(166,100):PRINT#1,SO$(3)
1840 YM=40
1850 IFSTICK(JO)=1THENYM=YM-20:FORT=1TO5
0:NEXTT
1860 IFYM=<40THENYM=40
1870 IFSTICK(JO)=5THENYM=YM+20:FORT=1TO5
0:NEXTT
1880 IFYM=>100THENYM=100
1890 PUT SPRITEO,(150,YM-3),1,0
1900 IFSTRIG(JO)=0THEN1850
1910 PUT SPRITEO,(0,209),0,0
1920 '
1930 'COMPROBACION RESPUESTAS
1940 LV=YM/20-1
1950 ON LV GOSUB1960,1990,2020,2050
1960 FORU=1TO3:FORA=1TO8
1970 IFSO$(U)=J1$(A)THEN2080
1980 NEXTA:NEXTU:GOTO 2100
1990 FORA=1TO8
2000 IFSO$(1)=J1$(A)THEN2110
2010 NEXTA:GOTO 2080
2020 FORA=1TO8
2030 IFSO$(2)=J1$(A)THEN2110
2040 NEXTA:GOTO 2080
2050 FORA=1TO8
2060 IFSO$(3)=J1$(A)THEN2110
2070 NEXTA:GOTO 2080
2080 LINE(152,118)-(245,130),11,BF:PSET(
155,120):COLOR 1:PRINT#1,"FIJATE BIEN":F
ORT=1TO250:NEXTT:LINE(152,118)-(245,130)
,12,BF:COLOR 15
2090 GOTO 1850
2100 GOTO 800
2110 IFH=2THENGOSUB2240:GOTO 1470
2120 IFH=3THENGOSUB2390:GOTO 240
2130 '
2140 'RUTINA ANOTA RESPUESTAS
2150 '
2160 'SI TIENE(3)LO QUE PIDIO(2)
2170 '
2180 FORU=1TO3:FORB=1TO9
2190 IFG2$(U,B)=J3$(A)THENG2$(U,B)="I":G
OTO 2210
2200 NEXTB:NEXTU
2210 Z(U)=Z(U)+1:IFZ(U)>8THENZ(U)=8
2220 RETURN
2230 '
2240 'SI TIENE(1) LO QUE PIDIO(2)
2250 '
2260 FORU=1TO3:FORB=1TO9
2270 IFG2$(U,B)=J1$(A)THENG2$(U,B)="I":G
OTO 2290
2280 NEXTB:NEXTU
2290 Z(U)=Z(U)+1:IFZ(U)>8THENZ(U)=8
2300 RETURN
2310 '
2320 ' NO TIENE NADIE LO QUE PIDE(2)
2330 '
2340 FORU=1TO3
2350 IFSO$(U)=C$(U)THENZ(U)=8
2360 NEXTU
2370 GOTO 1450
2380 '
2390 'SI TIENE(1)LO QUE PIDIO(3)
2400 '
2410 FORU=1TO3:FORB=1TO9
2420 IFG3$(U,B)=J1$(A)THENG3$(U,B)="I":G
OTO 2440
2430 NEXTB:NEXTU
2440 K(U)=K(U)+1:IFK(U)>8THENK(U)=8
2450 RETURN
2460 '
2470 'SI TIENE(2) LO QUE PIDIO(3)
2480 '
2490 FORU=1TO3:FORB=1TO9
2500 IFG3$(U,B)=J2$(A)THENG3$(U,B)="I":G
OTO 2520
2510 NEXTB:NEXTU
2520 K(U)=K(U)+1:IFK(U)>8THENK(U)=8
2530 RETURN
2540 '
2550 'NO TIENE NADIE LO QUE PIDE(3)
2560 '
2570 FORU=1TO3
2580 IFSO$(U)=C$(U)THENK(U)=8
2590 NEXTU
2600 GOTO 240
2610 '
2620 'ELIGE RUTA MAS CORTA A "C"
2630 '
2640 IFN=CTHENGOTO 2920
2650 IFN<CTHENGOTO 26  ELSE2670
2660 ON N GOTO 2680,2700,2720,2730,2750,
2770,2780,2790
2670 ON N GOTO 0,2800,2810,2820,2830,285
0,2870,2880,2900
2680 IFC=4ORC=7THENN=4ELSEN=2
2690 GOTO 2920
2700 IFC=3THENN=3ELSEN=5
2710 GOTO 2920
2720 N=6:GOTO 2920
2730 IFC=7THENN=7ELSEN=5
2740 GOTO 2920
2750 IFC=6THENN=6ELSEN=8
2760 GOTO 2920
2770 N=9:GOTO 2920
2780 N=8:GOTO 2920
2790 N=9:GOTO 2920
2800 N=1:GOTO 2920
2810 N=2:GOTO 2920
2820 N=1:GOTO 2920
2830 IFC=4THENN=4ELSEN=2
2840 GOTO 2920
2850 IFC=3THENN=3ELSEN=5
2860 GOTO 2920
2870 N=4:GOTO 2920
2880 IFC=7THENN=7ELSEN=5
2890 GOTO 2920
2900 IFC=6ORC=3THENN=6ELSEN=8
2910 GOTO 2920
2920 IFH=2THENN2=N:GOTO 1050
2930 IFH=3THENN3=N:GOTO 1520
2940 '
2950 '
2960 '
2970 '
2980 '
2990 'COMPRUEBA ACUSACION JUG.(1)
3000 '
3010 FORA=1TO3
3020 IFSO$(A)<>C$(A)THEN3050
3030 NEXTA
3040 GOTO 3350
3050 SCREENO,0,0:WIDTH36:COLOR 15,4,4
3060 A1$="T150SOM6000L8"
3070 A2$="O5C4C.C16.C4D+.D16.D.C16.C.O4B
16.O5C2"
3080 A3$="O4C4C.C16.C4D+.D16.D.C16.C.O3B
16.O4C2"
3090 PLAYA1$,A1$:PLAYA2$,A3$
3100 PLAYA3$
3110 FORA=5TO31:LOCATEA,1:PRINT"[":NEXT
3120 FORA=5TO31:LOCATEA,20:PRINT"[":NEXT
3130 FORA=1TO20:LOCATE5,A:PRINT"[":LOCAT
E31,A:PRINT"[":NEXT
3140 LOCATE9,3:PRINT"!ESTAS EQUIVOCADO!"
3150 LOCATE7,5:PRINT"EN REALIDAD EL CRIM
EN"
3160 LOCATE9,6:PRINT"FUE EJECUTADO EN:"
3170 LOCATE13,8:PRINTC$(1)
3180 LOCATE8,10:PRINT"EL ARMA EMPLEADA F
UE:"
3190 LOCATE13,12:PRINTC$(2)
3200 LOCATE11,14:PRINT"EL ASESINO FUE:"
3210 LOCATE13,16:PRINTC$(3)
3220 FORA=1TO2000:NEXTA
3230 FORA=6TO30:FORB=2TO19:LOCATEA,B:PRI
NTCHR$(32):NEXTB,A
3240 LOCATE7,4:PRINT"TU ACUSACION ES FAL
SA"
3250 LOCATE7,6:PRINT"Y POR LO TANTO QUED
AS"
3260 LOCATE7,8:PRINT"FUERA DE LA COMPETE
N-"
3270 LOCATE7,10:PRINT"CIA."
3280 LOCATE7,12:PRINT"LO LAMENTO."
3290 LOCATE8,14:PRINT"(OTRA PARTIDA?(S/N
)"
3300 PLAYA2$,A3$:PLAYA2$,A3$
3310 A$=INKEY$:IFA$=""THEN3310
3320 IF A$="S"ORA$="s"THENRUN160
3330 IFA$="N"ORA$="n"THENEND
3340 GOTO 3310
3350 '
3360 'FINAL
3370 SCREENO,0,0:COLOR 15,4,4
3380 A1$="T200SOM2000V12":PLAYA1$
3390 A2$="O5A2F+.G8AD2E8F+8G8A8GF+E2."
3400 A3$="O5F+2D.E8F2D2C+8D8E8F+8EDC+2"
3410 PLAYA2$:PLAYA3$
3420 FORA=5TO31:LOCATEA,1:PRINT"[":NEXT
3430 FORA=5TO31:LOCATEA,21:PRINT"[":NEXT
3440 FORA=1TO21:LOCATE5,A:PRINT"[":LOCAT
E31,A:PRINT"[":NEXT
3450 LOCATE12,4:PRINT"!!ES CIERTO!!"
3460 LOCATE7,6:PRINT"EL CRIMEN FUE EJECU
TADO"
3470 LOCATE13,8:PRINT"EN:";C$(1)
3480 LOCATE8,10:PRINT"EL ARMA EMPLEADA F
UE:"
3490 LOCATE13,12:PRINTC$(2)
3500 LOCATE11,14:PRINT"EL ASESINO FUE:"
3510 LOCATE13,16:PRINTC$(3)
3520 LOCATE9,18:PRINT"NO EXISTE EL CRIME
N"
3530 LOCATE14,19:PRINT"PERFECTO"
3540 FORA=1TO2000:NEXTA
3550 FORA=6TO30:FORB=2TO19:LOCATEA,B:PRI
NTCHR$(32):NEXTB,A
3560 PLAYA2$:PLAYA3$
3570 IFH=1THENLOCATE10,4:PRINT"TU SAGAZ
CEREBRO"
3580 IFH=2THENLOCATE10,4:PRINT"SHERLOCK
HOLMES"
3590 IFH=3THENLOCATE10,4:PRINT"HERCULES
POIROT"
3600 LOCATE8,8:PRINT"HA SIDO EL PRIMERO
EN"
3610 LOCATE8,10:PRINT"DESCUBRIR LA VERDA
D"
3620 LOCATE8,12:PRINT"EN ";Q;"MOVIMIENTO
S"
3630 LOCATE8,14:PRINT"!!FELICITACIONES!!
"
3640 LOCATE8,17:PRINT"(OTRA PARTIDA?(S/N
)"
3650 GOTO 3310
3660 '
3670 '******************************
3680 '*                            *
3690 '*      INSTRUCCIONES         *
3700 '*                            *
3710 '******************************
3720 '
3730 KEYOFF:SCREENO,0,0:WIDTH36:CLS
3740 LOCATE18,0:PRINT"INSTRUCCIONES"
3750 FORA=1TO36:PRINTCHR$(1)+CHR$(87);:N
EXT
3760 LOCATEO,2:PRINT"  En despiadada com
petencia con los  famosos detectives MIS
TER HOLMES y MONSIEUR  POIROT se debe es
clarecer el horrendo crimen de la MANSIO
N ADAMS averiguando:"
3770 PRINT:PRINT"1) La HABITACION del cr
imen"
3780 PRINT"2) El ARMA homicida"
3790 PRINT"3) El ASESINO."
3800 PRINT" Cada uno de los competidores
 tiene una lista con 8 diferentes coarta
dasque descartan la posible culpabildad
y que deberan comparar mediante sa-gaces
 preguntas."
3810 PRINT"Cada jugador a su turno avent
ura unatriple sospecha,dirigiendose a la
 HABITACION correspondiente siendo oblig
acion del jugador que sigue admitirsi fi
gura en su lista uno de los 3"
3820 PRINT"nombres,aclarando cual es,per
o solo al que le pregunt".PRESIONE SPACE
."
3830 A$=INKEY$:IFA$<>" "THEN3830
3840 CLS:PRINT"En caso de no tener ningu
no,la obli-gaci"n es del jugador siguien
te."
3850 PRINT"Al conocer la soluci"n y una
vez que se est  en la HABITACION que cor
responde se afirma la culpabilidad, qued
ando fuera del juego de no ser cierto."
3860 PRINT"Si tu memoria no es de elefan
te usa papel y lap/z."
3870 LOCATE11,13:PRINT"SELECCIONE"
3880 LOCATE11,15:PRINT"CURSOR"
3890 LOCATE11,17:PRINT"JOYSTICK(1)"
3900 LOCATE11,19:PRINT"JOYSTICK(2)"
3910 LOCATE10,21:PRINT"PRESIONE SPACE"
3920 Y=15:JO=0
3930 W=STICK(0):LOCATE9,Y:PRINT"  "
3940 IFW=1THENY=Y-2:IFY=<15THENY=15
3950 IFW=1THENJO=JO-1:IFJO=<0THENJO=0
3960 IFW=5THENY=Y+2:IFY=>19THENY=19
3970 IFW=5THENJO=JO+1:IFJO=>2THENJO=2
3980 LOCATE9,Y:PRINT"->"
3990 IFSTRIG(0)=0THENFORT=1TO100:NEXT:GO
TO 3930
4000 RETURN
4010 '******************************
4020 '*                            *
4030 '*   CARGA   DE   MATRICES    *
4040 '*                            *
4050 '******************************
4060 '
4070 DIML$(27),F$(3,9),G2$(3,9),G3$(3,9)
,RU(9,10):RESTORE5740
4080 FORA=1TO27
4090 READ D$
4100 L$(A)=D$
4110 NEXT A
4120 RESTORE5740
4130 FORA=1TO3:FORU=1TO9
4140 READD$
4150 F$(A,U)=D$:G2$(A,U)=D$:G3$(A,U)=D$
4160 NEXTU:NEXTA
4170 '
```

DISEÑO DE CARACTERES

Mostramos cómo una parte de la memoria almacena los caracteres. Su comprensión nos servirá cuando necesitemos diseñar alguna letra o símbolo en particular.

Las microcomputadoras que han adoptado la norma MSX en el mundo poseen una memoria RAM dedicada exclusivamente a almacenar los datos relativos a la pantalla. De tal modo TEXTOS (códigos de los caracteres y sus diseños), GRAFICOS (imagen de pantalla, colores y diseños) y SPRITES (diseños y atributos) son conservados en la denominada VRAM o Memoria RAM de VIDEO.

Esta memoria extra libera el resto de la RAM común de esa tarea, y agrega al sistema 16 Kbytes en la TALENT MSX 1 y nada menos que 128 Kbytes en la TALENT MSX 2 TURBO, llevándolas a un total de 80 Kbytes en el primer caso, y a 256 Kbytes en el segundo.

La memoria VRAM no es manejada directamente por el microprocesador Z80A, sino por un chip específico de video que recibe el nombre de VDP (Video Display Processor), el TMS 9919 en la MSX 1 y el TMS 9938A en la MSX 2.

El hecho de tener un chip dedicado al procesamiento de las salidas por pantalla no solo libra al Z80A de este pesado trabajo, sino que además permite un manejo de GRAFICOS y SPRITES de excelentes prestaciones y sumamente simple de realizar. Esto le otorga al equipamiento MSX especiales aptitudes para su uso en el ámbito educativo tanto en nivel primario con el uso de ACTORES en LOGO, cuanto en el secundario con los SPRITES del BASIC.

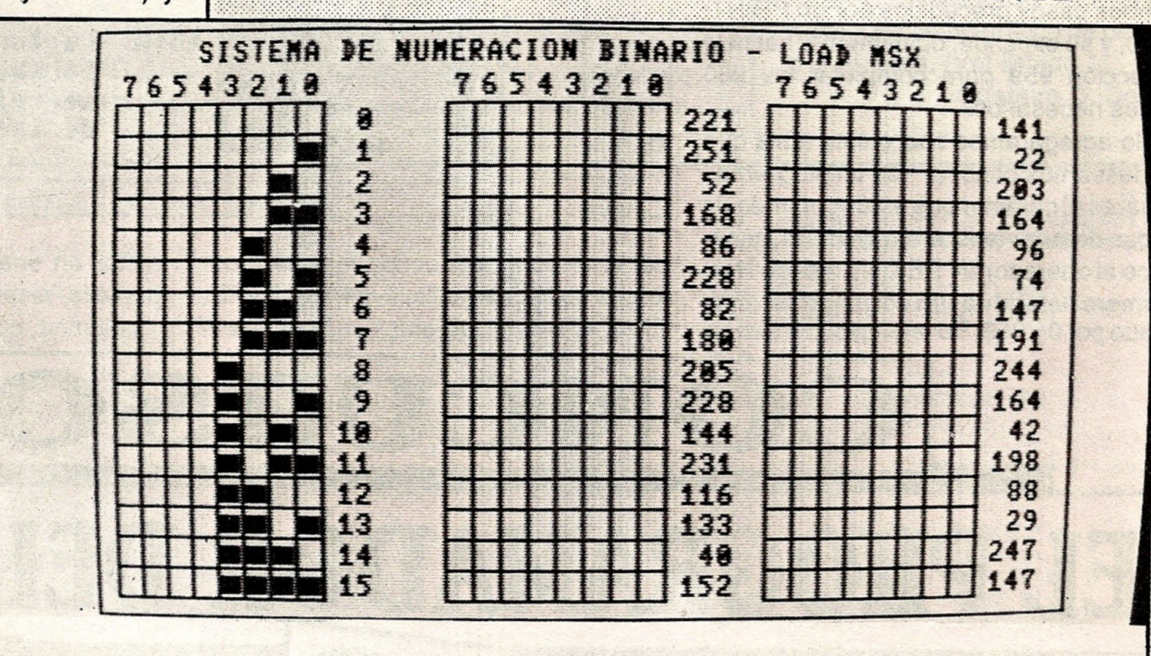
EL MODO 0 DE TEXTOS

En el modo 0 de textos (SCREEN 0), la TALENT puede manejar caracteres con dos colores (sobre un total de 16 en la MSX 1 o sobre 512 en la MSX 2), uno para el frente y otro para el fondo de la pantalla. La resolución máxima es de 40 columnas (u 80 en la MSX 2) por 24 filas, es decir que podemos llegar a ver hasta 40 caracteres en cada una de las 23 o 24 filas.

En este modo de pantalla no pueden trazarse gráficos con los comandos específicos (LINE, CIRCLE, PSET, etcétera) ni utilizarse SPRITES, aunque con un poco de imaginación pueden hacerse ciertos dibujos simples con lo caracteres predefinidos, o simular alguna movilidad cambiándolos de posición con LOCATEs y PRINTs.

La VRAM tiene dos zonas destinadas al manejo de este modo 0 de pantalla: a) la

UNA PROPUESTA EDUCATIVA



Apellido y nombre:...Dirección:...

A modo de desafío y con el objeto de desentumecer un poco las neuronas y practicar el sistema de numeración binario con que se generan los diseños de los caracteres, les ofrecemos la oportunidad de interactuar con nosotros completando y enviándonos una copia (si no quieren cortar la revista) del ejercicio que proponemos en la figura anexa. Entre las propuestas correctas sortearemos un libro. El cierre de la competencia es el 31/12/88.

La actividad consiste en completar con unos o simplemente pintar (como está en el ejemplo) los casilleros correspondientes al número BINARIO cuya equivalencia en DECIMAL figura a su lado.

Para aquellos que en forma habitual utilicen el rediseño de caracteres en sus programas, sugerimos preparar una tabla con los 256 patrones BINARIOS posibles para facilitar esa tarea. En el ejemplo presentado, ya tenemos los dieciséis primeros números.

En la parte operativa, recomendamos, al rediseñar una letra o símbolo, no perder de vista el hecho de que en el modo 0 de textos solo se utilizan las seis primeras columnas de cada carácter, de las cuales la sexta se reserva como separación, lo que produce un solapamiento de las dos últimas con las dos primeras del carácter siguiente.

Solo nos resta sugerirles que nos envíen por correo sus propios desarrollos sobre el tema para que podamos compartirlos con todos nuestros lectores, y así contribuir al engrandecimiento de la legión de usuarios MSX.

denominada TABLA DE IMAGEN DE PANTALLA y b) la TABLA DE GENERACION DE DISEÑOS o PATRONES DE CARACTERES.

TABLA DE IMAGEN DE PANTALLA

En esta tabla, que ya comentamos en un artículo en LOAD de octubre del 87 (Número 18), se almacenan los códigos ASCII de los 960 caracteres que constituyen la cuadrícula en que podemos dividir la pantalla (40 columnas x 24 filas).

La dirección de comienzo de la tabla puede conocerse por medio del comando BASE(0) o mediante VDP(2)*1024, en ambos casos precedidos por un PRINT; es 0, y se extiende, obviamente, hasta la dirección 959 para completar los 960 bytes necesarios.

Solo agregaremos sobre esta tabla que si deseamos conocer - en un programa - cuál es el carácter que está ocupando un lugar determinado en la pantalla, tenemos el operador VPEEK(N), siendo N un número natural dentro del rango determinado por 0 y 959. Por ejemplo, para conocer el carácter que ocupa la tercera columna de la cuarta fila: —> PRINT VPEEK(3*40+4). Si queremos colocar un carácter en una celda determinada de la pantalla, recurriremos a VPOKE fila * 40 + columna, código ASCII del carácter.

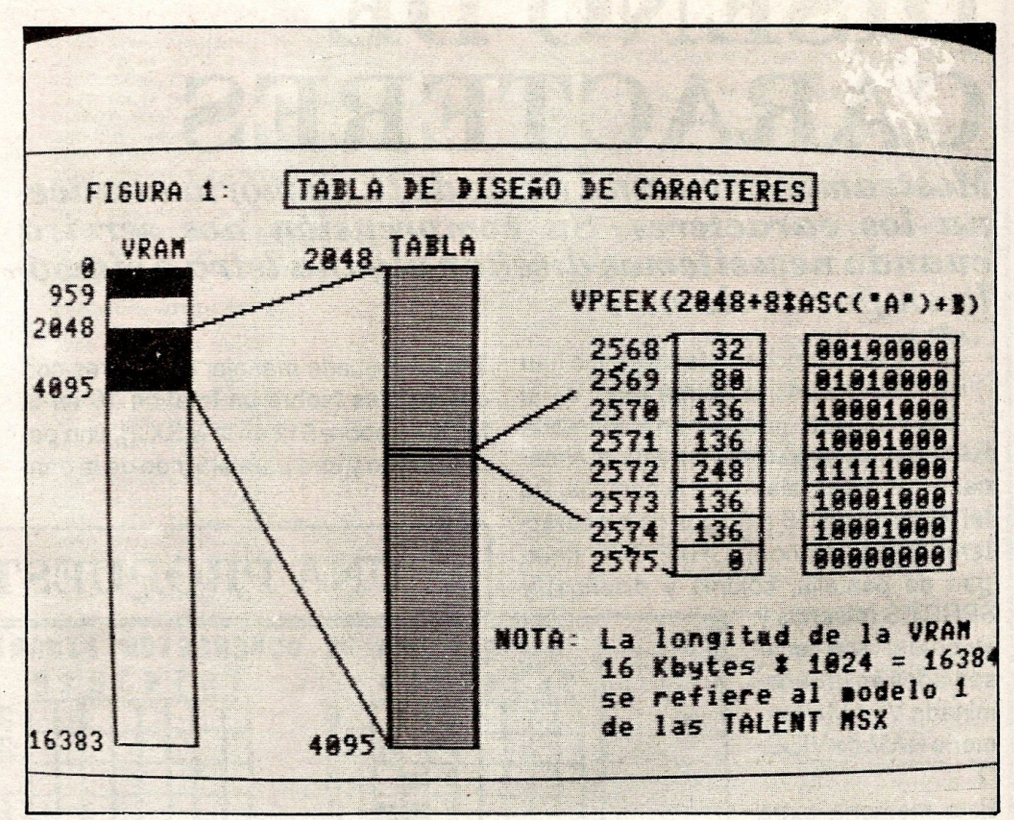


TABLA GENERADORA DE DISEÑOS O PATRONES DE LOS CARACTERES

En esta tabla de la VRAM, se almacenan los patrones de los 256 caracteres disponibles en el modo 0 de las MSX. Estos patrones son la información que el VDP necesita para 'encender' en la pantalla los pixeles (o puntos) que configuran el diseño con que identificamos cada una de las letras, números o símbolos.

Cada uno de los caracteres está definido por ocho bytes. Ello determina una longitud o espacio de memoria de 2.048 bytes (256 caracteres x 8 bytes cada uno). (Ver pantalla 1)

La dirección de comienzo de esta tabla podemos conocerla tipeando PRINT BASE (2) o bien PRINT VDP (4) *2048. Es 2048 en SCREEN 0, y se extiende hasta la dirección 4095 inclusive.

Para poder comprender mejor cómo nuestra amiga cibernética almacena un carácter (letra, número, símbolo o espacio vacío), imaginemos que cada letra es dibujada en una cuadrícula de 8 filas (bytes) por 8 columnas (bits). Cada conjunto de 8 bits (1 o 0) horizontales, pasados al sistema numérico decimal, constituye un byte. Ocho bytes consecutivos forman el patrón numérico con que la MSX 'conoce' un carácter. Poco romántico, ¿no?.

dirección	dec	binario
PRINT VPEEK (2568 + 0) =	32	00100000
PRINT VPEEK (2568 + 1) =	80	01010000
PRINT VPEEK (2568 + 2) =	136	10001000
PRINT VPEEK (2568 + 3) =	136	10001000
PRINT VPEEK (2568 + 4) =	248	11111000
PRINT VPEEK (2568 + 5) =	136	10001000
PRINT VPEEK (2568 + 6) =	136	10001000
PRINT VPEEK (2568 + 7) =	0	00000000

Para poder conocer los 8 bytes de un carácter en particular, debemos ubicar la dirección inicial de la Tabla de Diseños y aplicar sucesivos PRINT VPEEK(...), y para ello utilizamos la fórmula: 2048 + ASCII * 8. Para la letra 'A', como ejemplo, 2048 + 65 * 8 = 2568.

Notemos que la letra 'A' para nuestro equipo es la secuencia 32 80136 136 248 136 136 0. Y...¿cómo será que almacenamos "nosotros" el carácter 'A' en nuestro cerebro...?

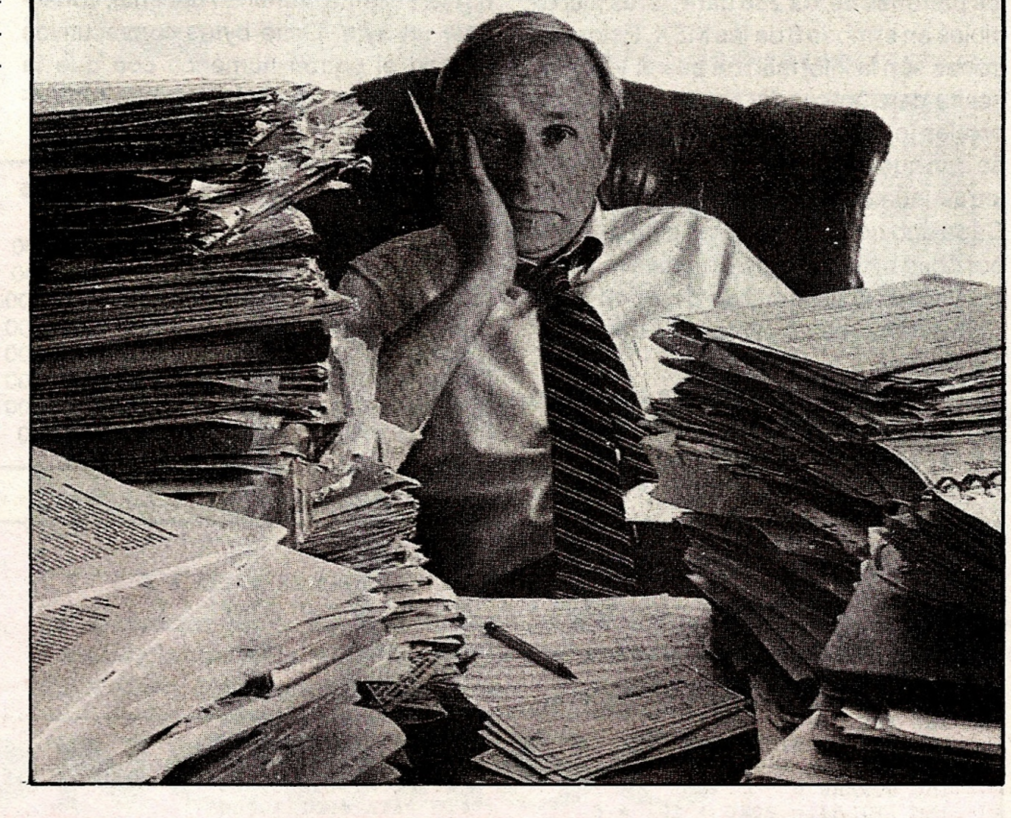
Para facilitar aún más la comprensión del tema y a modo de utilitario, preparamos un programa que nos da toda la información necesaria para cada carácter, incluyendo las direcciones de memoria, el valor almacenado en cada uno de los 8 bytes en los sistemas decimal y binario y el diseño gráfico de pantalla e impresora. (Ver listado 1)

Gustavo O. Delfino

```
100 REM ************************
105 REM *                      *
110 REM * DISENO DE CARACTERES *
115 REM *                      *
120 REM ************************
125 REM
130 REM  Por Gustavo O. Delfino
135 REM
140 REM * ASIGNACIONES *
145 REM   ------------
150 SCREEN 0
155 LET DI = 2048 + 191 * 8
160 FOR D = DI TO DI + 7
165   READ N
170   VPOKE D , N
175 NEXT D
180 DATA &b11111111
185 DATA &b10000000
190 DATA &b10000000
195 DATA &b10000000
200 DATA &b10000000
205 DATA &b10000000
210 DATA &b10000000
215 DATA &b11111111
220 REM
225 LET DI=2048 + 217*8
230 FOR D = DI TO DI + 7
235   READ N
240   VPOKE D , N
245 NEXT D
250 DATA &b10000000
255 DATA &b10000000
260 DATA &b10000000
265 DATA &b10000000
270 DATA &b10000000
275 DATA &b10000000
280 DATA &b10000000
285 DATA &b10000000
290 REM
295 LET DI=2048 + 155 * 8
300 FOR D = DI TO DI + 7
305   READ N
310   VPOKE D , N
315 NEXT D
320 DATA &b00000000
325 DATA &b01110000
330 DATA &b11111000
335 DATA &b11111000
340 DATA &b11111000
345 DATA &b01110000
350 DATA &b00000000
355 DATA &b00000000
360 REM
365 LET DI=2048 + 152 * 8
370 FOR D = DI TO DI + 7
375   READ N
380   VPOKE D , N
385 NEXT D
390 DATA &b00000000
395 DATA &b01110000
400 DATA &b10001000
405 DATA &b10001000
410 DATA &b10001000
415 DATA &b01110000
420 DATA &b00000000
425 DATA &b00000000
430 REM
435 LET DI=2048 + 158 * 8
440 FOR D = DI TO DI + 7
445   READ N
450   VPOKE D , N
455 NEXT D
460 DATA &b00110000
465 DATA &b01001000
470 DATA &b01001000
475 DATA &b11111100
480 DATA &b01001000
485 DATA &b11111100
490 DATA &b01001000
495 DATA &b00000000
500 REM
505 REM * ACONDICIONAMIENTO *
510 REM   -----------------
515 WIDTH 40
520 KEY OFF
525 COLOR 1 , 3
530 REM
535 REM * RUTINA OPERATIVA *
540 REM   ----------------
545 CLS
550 LOCATE 1,2 : PRINT "Pulse
    cualquier tecla:";
555 LOCATE 26,2;1
560 LET K$ = INKEY$ : IF K$ =""
    THEN GOTO 560 ELSE LOCATE,,0
565 IF ASC(K$)<32 OR ASC(K$)>255
    THEN GOTO 555 ELSE PRINT K$
570 LET DI = 2048 + ASC(K$) * 8
575 LOCATE 0,3:PRINT "XWRWWWRWWW
  WWWWWWWWWWWWWWWWWWWWWWWWWWWWWY
580 LOCATE 0,4:PRINT "VcVASCV Pa
  trón en decimal (8 bytes)    V
585 LOCATE 0,5:PRINT "TWUWWWUWWW
  WWWWWWWWWWWWWWWWWWWWWWWWWWWWWS
590 LOCATE 0,6:PRINT "V V   V
                               V
595 LOCATE 0,7:PRINT "ZWQWWWQWWW
  WWWWWWWWWWWWWWWWWWWWWWWWWWWWW;
600 REM
605 LOCATE 1,6 : PRINT K$ ;
610 LOCATE 3 : PRINT USING "###"
    ; ASC(K$) ;
615 LOCATE 7
620 FOR B = 0 TO 7
625   LET C = VPEEK(DI+B)
630   PRINT STR$(C) ;
635 NEXT B
640 LOCATE 0,9:PRINT " b dir  de
  c binario  pantalla impresora"
645 FOR B = 0 TO 7
650   LET F$ = "00000000"
655   LET C = VPEEK(DI+B)
660   LOCATE 0,11+B : PRINT B ;
665   LOCATE 2 : PRINT DI+B ;
670   LOCATE 8 : PRINT USING
      "###" ; C ;
675   MID$(F$,9-LEN(BIN$(C))) =
      BIN$(C)
680   LOCATE 12 : PRINT F$ ;
685   LOCATE 21
690   FOR BIT = 1 TO 8
695     IF MID$(F$,BIT,1) = "0"
        THEN PRINT CHR$(191) ;
        ELSE PRINT CHR$(200) ;
700   NEXT BIT
705   PRINT CHR$(217) ;
710   LOCATE 30
715   FOR BIT = 1 TO 8
720     IF MID$(F$,BIT,1) = "0"
        THEN PRINT CHR$(152) ;
        ELSE PRINT CHR$(155) ;
725   NEXT BIT
730 NEXT B
735 LOCATE 0,20:PRINT "Pulse cua
  lquier tecla para continuar"
740 LOCATE 6,21 : PRINT "O 'ESC'
  para finalizar"
745 LET K$ = INKEY$ : IF K$ = ""
  THEN GOTO 745
750 IF NOT K$ = CHR$(27) THEN
  GOTO 545
755 REM
760 REM * RESTAURACION *
765 REM   ------------
770 CLS
775 KEY ON
780 COLOR 15 , 4
785 END
```

INTERCAMBIANDO ARCHIVOS ENTRE EL DBASE Y EL MSX BASIC.

En este caso, se trata de un utilitario que permite leer archivos generados por el dBASE II (programa que se ha descripto en esta revista con bastante detalle) desde el MSX BASIC, e incluso generar un programita de ejemplo que arma los campos del archivo y realiza un ciclo para mostrar todos los registros. A partir del programa generado se podrán armar programas más sofisticados para el intercambio de la información entre BASIC y dBASE.



UN POCO DE TEORIA

Los archivos dBASE II generados tanto en MSX como en las IBM PC compatibles se arman de una forma estándar. Supongamos que tenemos un archivo llamado SOFT.DBF. Si hacemos un dump del mismo veremos lo que indica la figura 1 (en los primeros 500 bytes).

En la primera columna se indica la posición relativa dentro del archivo en hexadecimal. En las 8 siguientes, el código ASCII (también en hexadecimal) de los caracteres que componen esa línea y en la última, se muestran los caracteres propiamente dichos,



FIGURA 1

```
A:SOFT.DBF:                                    0038: 4E 55 4D 45 52 4F 00 00  NUMERO..
0000: 02 53 00 0A 08 58 64 00  .S...Xd.        0040: 00 00 00 43 03 09 71 00  ...C..q.
0008: 54 49 54 55 4C 4F 00 00  TITULO..        0048: 4D 53 58 00 00 00 00 00  MSX.....
0010: 00 00 00 43 23 B9 70 00  ...C¿9p.        0050: 00 00 00 43 02 0C 71 00  ...C..q.
0018: 54 49 50 4F 00 00 00 00  TIPO....        0058: 4D 53 58 3A 38 30 00 00  MSX:80..
0020: 00 00 00 43 19 DC 70 00  ...C.Ñp.        0060: 00 00 00 43 02 0E 71 00  ...C..q.
0028: 43 41 4A 41 00 00 00 00  CAJA....        0068: 4D 53 58 32 00 00 00 00  MSX2....
0030: 00 00 00 43 14 F5 70 00  ...C.up.        0070: 00 00 00 43 02 10 71 00  ...C..q.
```

```
0078: 4D 41 4E 55 41 4C 00 00  MANUAL..  0198: 00 00 00 00 00 00 00 00  ........
0080: 00 00 00 4C 01 12 71 00  ...L..q.  01A0: 00 00 00 00 00 00 00 00  ........
0088: 43 41 4E 54 49 44 41 44  CANTIDAD  01A8: 00 00 00 00 00 00 00 00  ........
0090: 00 00 00 4E 03 13 71 00  ...N..q.  01B0: 00 00 00 00 00 00 00 00  ........
0098: 56 41 4C 4F 52 00 00 00  VALOR...  01B8: 00 00 00 00 00 00 00 00  ........
00A0: 00 00 00 4E 06 16 71 02  ...N..q.  01C0: 00 00 00 00 00 00 00 00  ........
00A8: 0D 00 00 00 00 00 00 00  ........  01C8: 00 00 00 00 00 00 00 00  ........
00B0: 00 00 00 00 00 00 00 00  ........  01D0: 00 00 00 00 00 00 00 00  ........
00B8: 00 00 00 00 00 00 00 00  ........  01D8: 00 00 00 00 00 00 00 00  ........
00C0: 00 00 00 00 00 00 00 00  ........  01E0: 00 00 00 00 00 00 00 00  ........
00C8: 00 00 00 00 00 00 00 00  ........  01E8: 00 00 00 00 00 00 00 00  ........
00D0: 00 00 00 00 00 00 00 00  ........  01F0: 00 00 00 00 00 00 00 00  ........
00D8: 00 00 00 00 00 00 00 00  ........  01F8: 00 00 00 00 00 00 00 00  ........
00E0: 00 00 00 00 00 00 00 00  ........  0200: 00 00 00 00 00 00 00 00  ........
00E8: 00 00 00 00 00 00 00 00  ........  0208: 00 20 41 42 53 54 41 54  . ABSTAT
00F0: 00 00 00 00 00 00 00 00  ........  0210: 20 20 20 20 20 20 20 20
00F8: 00 00 00 00 00 00 00 00  ........  0218: 20 20 20 20 20 20 20 20
0100: 00 00 00 00 00 00 00 00  ........  0220: 20 20 20 20 20 20 20 20
0108: 00 00 00 00 00 00 00 00  ........  0228: 20 20 20 20 20 55 54 49       UTI
0110: 00 00 00 00 00 00 00 00  ........  0230: 4C 49 54 41 52 49 4F 20  LITARIO
0118: 00 00 00 00 00 00 00 00  ........  0238: 20 20 20 20 20 20 20 20
0120: 00 00 00 00 00 00 00 00  ........  0240: 20 20 20 20 20 20 55 54        UT
0128: 00 00 00 00 00 00 00 00  ........  0248: 49 4C 49 54 41 52 49 4F  ILITARIO
0130: 00 00 00 00 00 00 00 00  ........  0250: 53 20 20 20 20 20 20 20  S
0138: 00 00 00 00 00 00 00 00  ........  0258: 20 20 30 31 31 2E 2E 53    011..S
0140: 00 00 00 00 00 00 00 00  ........  0260: 49 53 49 46 20 20 30 20  ISIF  0
0148: 00 00 00 00 00 00 00 00  ........  0268: 20 30 2E 30 30 20 42 41   0.00 BA
0150: 00 00 00 00 00 00 00 00  ........  0270: 53 49 43 20 38 30 20 43  SIC 80 C
0158: 00 00 00 00 00 00 00 00  ........  0278: 4F 4D 50 49 4C 45 52 20  OMPILER
0160: 00 00 00 00 00 00 00 00  ........  0280: 20 20 20 20 20 20 20 20
0168: 00 00 00 00 00 00 00 00  ........  0288: 20 20 20 20 20 20 20 20
0170: 00 00 00 00 00 00 00 00  ........  0290: 20 4C 45 4E 47 55 41 4A   LENGUAJ
0178: 00 00 00 00 00 00 00 00  ........  0298: 45 20 20 20 20 20 20 20  E
0180: 00 00 00 00 00 00 00 00  ........
0188: 00 00 00 00 00 00 00 00  ........
0190: 00 00 00 00 00 00 00 00  ........
```

si su código ASCII supera el valor &H1F.

La información se da de la forma que indica la figura 2 (tomando como primer byte el número 0)

Los siguientes bytes describen cada campo. (ver figura 3)

Se admiten hasta 32 campos por registro. Para indicar el fin de los campos, la posición que correspondería al campo siguiente tiene como primer carácter el CR (ASCII 13). Finalmente, los datos de la base se almacenan de corrido en forma similar al MSX BASIC con archivos relativos, en ASCII.

El área de datos se i-

FIGURA 2

Byte	Indica
00	Código 02:Indica archivo dBASE
01-02	Cantidad de registros
03	Dia última actualización
04	Mes última actualización
05	Año última actualización
06-07	Cantidad de caracteres por registro + 1

FIGURA 3

Posición relativa	Descripción
0 an 9	Nombre campo (se completa con el carácter ASCII 0 hasta llegar a 10 caracteres)
10	Carácter ASCII 0
11	Tipo (en ASCII: "C" para carácter, "N" para numérico, "L" para lógico)
12	Longitud (hasta 254)
13-14	Dirección de memoria donde se almacena en dBASE
15	Cantidad de decimales

nicia en la posición &H20A dentro del archivo.
Para intercambiar datos con el BASIC se usa la siguiente secuencia:
Siguiendo con el ejemplo del archivo SOFT.DBF y estando en dBASE, ingresemos lo siguiente:

USE SOFT
COPY TO SOFT SDF
lo que generará el archivo SOFT.TXT con todos los datos en ASCII, separando cada registro con la secuencia CR/LF (carácter ASCII13/carácter ASCII 10).
El programa que se incluye en este artículo es un pequeño utilitario que permite tomar el archivo SOFT.DBF y analizar su estructura aplicando lo visto, así como generar un programa BASIC que leerá el archivo SOFT.TXT (o el que se indique) y lo imprimirá en pantalla. A partir del programa generado (que si se opta por grabarlo en disco, se podrá cargar con un simple LOAD), se deja librada a la imaginación del lector la aplicación de intercambio de datos que se le quiera dar al susodicho programita.
Cuando se hace el listado final del programa generado, usamos el "truco" del Prof. Gustavo Delfino que permite preservar los últimos renglones de la pantalla contra el "scroll".

```
10 ' *
20 ' * Analizador dBASE II
30 ' *  Revista: Load MSX
40 ' *  Rincon del usuario Talent
50 ' *  Autor: Hugo Caro
60 ' *
70 CLEAR 1024:CLS:WIDTH40:KEYOFF:PRINT "Analizador de archivos dBASE II"
80 DEF FNCERO$(NU,CU)=RIGHT$(STRING$(CU,"0")+RIGHT$(STR$(NU),LEN(STR$(NU))-1),CU
):DIM CA$(32,4),EQ$(32),PB$(38),E2$(32),FL%(32)
90 '
100 ' * Menu principal
110 '
120 LOCATE 0,5:PRINT TAB(5);"1) Muestra estructura
130 PRINT TAB(5);"2) Muestra y genera programa BASIC
140 PRINT TAB(5);"3) Fin
150 LOCATE 0,22:INPUT "Su opcion (1-3):";OP%
160 IF OP%<1 OR OP%>3 THEN 70
170 ON OP% GOTO 230,230,210
180 '
190 ' * Fin programa
200 '
210 CLS:PRINT"Adios...":END
220 '
230 ' * Analizador estructura dBASE
240 '
250 CLS:PRINT:PRINT "Nombre del Archivo:":INPUTAR$
260 ON ERROR GOTO 1300
270 OPEN AR$ FOR INPUT AS 1:CLOSE 'Chequea si existe el archivo
280 OPEN AR$ AS 1 LEN=1
290 FIELD 1,1 AS A$
300 GET#1,1:IF A$<>CHR$(2) THEN PRINT "No es un archivo dBASE":CLOSE:END
310 CLS
320 PRINT "Estructura del archivo:  ";AR$
330 GET#1:LO=ASC(A$):GET#1:HI=ASC(A$)
340 NR=LO+256*HI
350 PRINT "Numero de Registros: ";FNCERO$(NR,5)
360 GET#1:DI=ASC(A$):GET#1:ME=ASC(A$):GET#1:AN=ASC(A$)
370 PRINT "Fecha de ultima actualizacion: ";FNCERO$(DI,2);"/";FNCERO$(ME,2);"/";
FNCERO$(AN,2):GET#1:LO=ASC(A$):GET#1:HI=ASC(A$):TT=LO+256*HI
380 PRINT"Longitud Registro: ";FNCERO$(TT-1,5):PRINT:PRINT "CMP       NOMBRE
TIPO LONG      DEC":NC=0
390 GOSUB 1100:IF ASC(NO$)=13 THEN 450
400 NC=NC+1:CA$(NC,1)=NO$:PRINT FNCERO$(NC,3);TAB(8);CA$(NC,1);
410 GET#1:GET#1:CA$(NC,2)=A$:PRINTTAB(21);CA$(NC,2);
420 GET#1:CA$(NC,3)=FNCERO$(ASC(A$),3):PRINT TAB(26);CA$(NC,3);
430 GET#1:GET#1:GET#1:CA$(NC,4)=FNCERO$(ASC(A$),3):IF ASC(A$)<>0 THEN PRINT TAB(
33);CA$(NC,4) ELSE PRINT
440 GOTO 390
450 PRINT "** TOTAL **";TAB(24);FNCERO$(TT,5):TT=TT-1:PRINT:GOSUB 1060:CLOSE
460 IF OP%=1 THEN 70
```

```
470 '
480 ' * Generar programa BASIC
490 '
500 CLS:PRINT "Generando programa BASIC..."
510 XX=INSTR(1,AR$,"."):IF XX THEN AR$=LEFT$(AR$,XX-1):PRINT:PRINT"Programa: ";A
R$+".BAS":PRINT
520 PRINT:PRINT "Generando encabezado...";
530 A$="OPEN "+CHR$(34)+AR$+".TXT"+CHR$(34)+" AS #1 LEN="+STR$(TT+2):NL=0:NP=0
540 GOSUB 1180:PRINT "Completo"
550 '
560 '* Analizar variables (por compatibilidad BASIC)
570 '
580 PRINT "Analizando variables:":PRINT " Fase I...";
590 FOR I=1 TO NC
600 EQ$(I)=LEFT$(CA$(I,1),2):FL%(I)=VAL(CA$(I,3))
610  XX=INSTR(1,EQ$(I),":"):IF XX THEN EQ$(I)=LEFT$(EQ$(I),XX-1)+RIGHT$(EQ$(I),L
EN(EQ$(I))-XX)
620 NEXT I:PRINT"Completa"
630 TP=0:PRINT " Fase II...";
640 TP=TP+1:IF TP>NC THEN PRINT"Completa":GOTO720
650 CC=0:CH=1:BF=0
660 IF EQ$(TP)="IF" OR EQ$(TP)="FN" OR EQ$(TP)="OR" OR EQ$(TP)="ON" OR EQ$(TP)="
TO" THEN OO=1:GOTO 700
670 OO=0:FOR J=1 TO TP
680  IF E2$(J)=EQ$(TP) THEN OO=1:GOTO700
690 NEXT J
700 IF OO THEN GOSUB 1220:GOTO 660
710 E2$(TP)=EQ$(TP):GOTO 640
720 '
730 ' * Construye linea de FIELD
740 '
750 PRINT "Construyendo linea FIELD...";
760 A$="FIELD #1,"
770 FOR I=1 TO NC
780  A$=A$+STR$(FL%(I))+" AS "+E2$(I)+"$":IF I<NC THEN A$=A$+","
790 NEXT I:GOSUB1180:PRINT "Completo"
800 IF NR=0 THEN A$="' No hay registros":GOSUB1180:PRINT"No hay registros...":GO
TO 880
810 A$="FOR I%=1 TO"+STR$(NR):GOSUB1180
820 A$="GET#1:PRINT ":PRINT "Construyendo ciclo lectura...";
830 FOR I=1 TO NC:A$=A$+E2$(I)+"$"
840 IF I<NC THEN A$=A$+";"
850 NEXT I:PRINT "Completo"
860 GOSUB 1180
870 A$="NEXT I%:CLOSE#1:END":GOSUB 1180
880 PRINT "Construyendo tabla equivalencias..."
890 NL=90:A$="' Tabla de equivalencias":GOSUB 1180
900 FOR I=1 TO NC
910 A$="' * "+E2$(I)+"$ = "+CA$(I,1):GOSUB 1180
920 NEXT I
930 '
940 CLS:PRINT "Graba el programa? (S/N)";
950 A$=INPUT$(1):PRINTA$:IF INSTR(1,"SsNn",A$)=0 THEN 940
960 IF A$="N" OR A$="n" THEN SN=0 ELSE SN=1
970 IF SN THEN OPEN AR$+".BAS" FOR OUTPUT AS #1
980 PO=PEEK(&HF3B1):POKE&HF3B1,22:FOR I=1 TO NP
990  PRINT PB$(I):IF SN THEN PRINT#1,PB$(I)
1000 NEXT I:IF SN THEN CLOSE#1
1010 GOSUB 1060:POKE&HF3B1,PO
1020 GOTO 70
1030 '
1040 '* Pausa
1050 '
1060 LOCATE 0,22:PRINT "Pulse una tecla...";:A$=INPUT$(1):LOCATE 0,22:PRINTCHR$(
27)"K":RETURN
1070 '
1080 ' * Obtiene nombre campo
```

```
1090 '
1100 NO$="":FOR I=1 TO 10
1110 GET#1:IF ASC(A$)=0 THEN 1130
1120 NO$=NO$+A$
1130 NEXT I
1140 RETURN
1150 '
1160 '* Genera Linea BASIC
1170 '
1180 NL=NL+10:NP=NP+1:PB$(NP)=STR$(NL)+" "+A$:RETURN
1190 '
1200 ' * Analizador de variables BASIC
1210 '
1220 IF BF THEN 1250
1230 CH=CH+1:IF CH>LEN(CA$(TP,1))THEN BF=1:LC=47:GOTO 1250
1240 EQ$(TP)=LEFT$(EQ$(TP),1)+MID$(CA$(TP,1),CH,1):RETURN
1250 LC=LC+1:IF LC=58 THEN LC=65
1260 EQ$(TP)=LEFT$(EQ$(TP),1)+CHR$(LC):RETURN
1270 '
1280 ' * Rutinas de ERROR
1290 '
1300 IF ERL=270 THEN PRINT:PRINT "No existe ese archivo!":GOSUB 1060:RESUME 70
1310 PRINT "Error:";ERR;"Linea:";ERL:ON ERROR GOTO 0:END
```

ENVIANDO MENSAJES ALMACENADOS EN UN ARCHIVO EN ACAMATICA

Los usuarios del modem Talent TMX-510 generalmente se comunican con ACAmática y cuando entran a dicha base de datos comienzan a intercambiar información con otros socios o nos consultan a INFOTALENT en ACAmática, al número T10003.
Cuando se utiliza este sistema y se quiere enviar un mensaje, se puede optar por dos caminos:
1) Ingresar desde el teclado el mensaje en el momento en que ACAmática lo solicita.
2) Enviar un archivo previamente almacenado en disco o casete.
El primer camino requiere saber dactilografía, si no se desea perder demasiado tiempo en línea, y se trabaja como una máquina de escribir: si me equivoqué en algo y me doy cuenta luego de pulsar <RETURN>..., deberé dejarlo así o abortar el mensaje.
El segundo camino, con el soft que incorpora el modem TMX-510 no se puede realizar directamente, pues no prevé que el Host (la computadora que maneja la BBS) tenga alguna demora en ir aceptando cada línea; lo que hace que ACAmática reciba la cuarta parte del mensaje enviado.
Es por esto que incluimos a continuación una versión modificada del programa modem para Talent en su parte BASIC (ya que aunque está en ROM, el programa terminal está en BASIC y en ASSEMBLER) que se puede usar con ACAmática.
La forma de utilizarlo es la siguiente:
1) Estando en MSX BASIC, ingresemos el clásico:
CALL MENU
2) Una vez que nos encuentremos en el menú, pulsemos <CTRL><STOP>
3) Carguemos el programa nuevo y ejecutémoslo.
Ya estaremos en condiciones de enviar un archivo en ACAmail de la forma habi-



tual, teniendo en cuenta que:
1) Se irá enviando un renglón por vez
2) Deberemos pulsar una tecla para que continúe el envío (o <ESC> para cancelarlo)
3) El archivo deberá tener un CR (Retorno de carro, ASCII 13) cada 38 caracteres o menos
4) No deberá excederse de los 10 renglones.

Esperemos que esta nueva versión sea de utilidad para lo"ACAmáticos".

```
560 DEFINT A-Z
570 CLS:KEYOFF:LOCATE,,Ø
580 B(Ø)=Ø:B(1)=1
650 DEFUSR=&HDØØ9:DEFUSR3=&HDØØ3
710 IF PEEK(&HFDØA)=&H33THEN I=USR3(1)
720 IF PEEK(&HFDØA)<>&H99THEN POKE&HFDØA,&H99
730 A$(1)="->"+CHR$(29)+CHR$(29):A$(Ø)="  "+CHR$(29)+CHR$(29):IT=1
740 SCREENØ:WIDTH40:LOCATEØ,Ø:PRINT TAB(11);"MODEM TALENT MSX          v.3.2":PRINT
TAB(12);"MENU PRINCIPAL":FOR I=1TO40:PRINT"-";:NEXT
750 LOCATE5,10:PRINT"BASIC"
760 LOCATE5,11:PRINT"ACAMATICA, DELPHI Y OTROS"
770 IO=3:GOSUB1000:IF OK=1THEN PRINTTAB(5);"PROCESADOR DE TEXTO":IT=IT+1:B(IT)=I
O
780 IO=9:GOSUB1000:IF OK=1THEN PRINTTAB(5);"PLANILLA DE CALCULO":IT=IT+1:B(IT)=I
O
790 LOCATE3,10:PRINT A$(Ø);
800 A$=INKEY$
810 IF K>17THEN K=Ø:N=1-N:LOCATE3,10+A:PRINT A$(N);
820 IF A$=""THEN K=K+1:GOTO800
830 IF ASC(A$)=30THEN870
840 IF ASC(A$)=31THEN910
850 IF ASC(A$)=13THEN950
860 GOTO800
870 LOCATE3,10+A:PRINT A$(Ø);
880 A=A-1:IF A<ØTHEN A=IT
890 LOCATE3,10+A:PRINT A$(1);
900 GOTO800
910 LOCATE3,10+A:PRINT A$(Ø);
920 A=A+1:IF A>IT THEN A=Ø
930 LOCATE3,10+A:PRINT A$(1);
940 GOTO800
950 OUT&H8D,Ø:I=USR3(Ø):I=USR3(2):OUT(&H8D),B(A)
960 IF B(A)=9THEN I=USR3(2):CALL MSXPLAN
970 IF B(A)=3THEN I=USR3(2):CALL WRITE
980 IF B(A)=ØTHEN I=USR3(Ø):CLS:NEW
990 IF B(A)=1THEN OUT&H8D,Ø:I=USR3(Ø):RUN1070
1000 OUT(&H8D),IO
1010 D=USR(&H4000)
1020 IF HEX$(D)="41"THEN OK=1ELSE OK=Ø
1030 RETURN
1070 CLEAR1024,&HCFØØ:DEFSNG A-Z
1080 MAXFILES=2:ON ERROR GOTO3220
1090 CSI=&HDØ18:GSI=&HDØ16:PTR=&HDØ12:PR=&HDØ1A
1100 DEFUSR=&HDØØ6:DEFUSR2=&HDØØØ:DEFUSR4=&HDØØC:DEFUSR5=&HDØØF
1140 CLS:KEY OFF
1150 DIM MA$(3,6),PR$(3,6),MX%(6),OP%(6)
1160 RESTORE3160:FOR I%=ØTO6:READ MX(I%),OP(I%),CL(I%):NEXT I%:CUR$(1)="->":CUR$
(Ø)="  "
1170 FOR I%=ØTO6:FOR J%=ØTO MX(I%):READ MA$(J%,I%)
1180 PR$(J%,I%)=MA$(J%,I%)
1190 IF OP(I%)=J%THEN A$=MA$(J%,I%):GOSUB2560:PR$(J%,I%)=B$
1200 NEXT J%:NEXT I%
1210 FOR I=1TO10:READ A$:KEY I,A$:NEXT
1220 KEY2,STR$(&H7FFF)
1230 IF PEEK(&HFAFC)THEN KEY9,"Ancho"
1270 EF%=2:CR%=Ø:EC%=Ø:OP%=Ø:NP%=Ø:WI=40:POKE PTR,Ø:POKE PTR+1,Ø:POKE PTR+2,Ø:PO
KE PTR+3,Ø:POKE GSI,Ø:POKE CSI,1:POKE PR,Ø:GOSUB2410
1280 GOSUB3800:OPEN"comØ:"AS1
1320 A$=INKEY$
1330 IF A$=CHR$(24)THEN3380
1340 IF LOC(1)THEN GOSUB1570
1350 IF A$=""THEN1320
```

```
1390 IF CR%AND A$=CHR$(13)THEN A$=A$+CHR$(10)
1400 IF OP(0)<>1THEN PRINT#1,A$;:GOTO1500
1430 IF(INP(&H82)AND1)=0THEN1320
1440 IF CM%=0THEN_COMDTR(,1):CM%=1
1450 PRINT#1,A$;
1460 IF INSTR(1,A$,CHR$(13))THEN_COMDTR(,0):CM%=0
1500 IF EC%THEN A$=USR2(A$):GOSUB1590
1510 GOTO1320
1570 B$=INPUT$(LOC(1),1)
1580 B$=USR2(B$)
1590 IF PEEK(GSI)THEN FR=PEEK(PTR)+PEEK(PTR+1)*256:IF FR<&H7FFF THEN KEY2,STR$(&
H7FFF-FR)ELSE OK%=1:KEY2,"0000":CLOSE2:GOSUB2730
1600 RETURN
1640 IF PEEK(PR)=0AND(INP(&H90)AND2)<>2THEN KEY1,"Troff":BEEP:POKE PR,1ELSE POKE
 PR,0:KEY1," Tron"
1650 RETURN
1690 RETURN
1730 IF PEEK(GSI)OR(PEEK(PTR+1)=&H80)THEN POKE GSI,0:KEY3,"Graba":BEEP:RETURN
1740 POKE GSI,1:KEY3,"No Graba":RETURN
1780 CLS:RETURN
1820 IF EC%=0THEN KEY6,"Eco:Si":EC%=1ELSE KEY6,"Eco:No":EC%=0
1830 RETURN
1840 IF CR%=0THEN KEY7,"LF:Si":CR%=1ELSE KEY7,"LF:No":CR%=0
1850 RETURN
1900 CLOSE:CALL COMINI("8n1n",300,300):WIDTH40:LOCATE,,0:FOR I%=1TO9:KEY(I%)OFF:
NEXT:A=USR(0):KEYOFF:LOCATE,,0
1910 PRINTTAB(14);"Parametros"
1920 PRINT
1930 PRINT"Velocidad | Norma  |    Modo     | Pari"
1940 PRINT"          |        |             | dad "
1950 PRINT"──────────┼────────┼─────────────┼─────"
1960 PRINT"          |        |             |     "
1970 PRINT"          |        |             |     "
1980 PRINT"          |        |             |     "
1990 PRINT"──────────┴────────┴─────────────┴─────"
2000 PRINT
2010 PRINT"       ┌──────┐  ┌──────┐  ┌──────┐"
2020 PRINT"       | Xon/ |  | Long.|  | Stop |"
2030 PRINT"       | Xoff |  | Datos|  | Bits |"
2040 PRINT"       ├──────┤  ├──────┤  ├──────┤"
2050 PRINT"       |      |  |      |  |      |"
2060 PRINT"       |      |  |      |  |      |"
2070 PRINT"       └──────┘  |      |  |      |"
2080 PRINT"                 |      |  └──────┘"
2090 PRINT"                 └──────┘"
2100 LOCATE0,21:PRINT"Utilice: Cursores para cambiar de OPCION";:PRINT TAB(9);"R
ETURN para seleccionar":PRINT TAB(9);"ESC para salir";
2110 GOSUB2250:X=0:Y=5:P=0
2120 A$=INKEY$:IF A$=""THEN T%=T%+1:IF T%=15THEN T%=0:LOCATE X,Y:PRINT CUR$(TI%)
;:TI%=1-TI%:GOTO2120ELSE2120
2130 LOCATE X,Y:PRINT CUR$(0)
2140 IF A$=CHR$(27)THEN WIDTH WI:GOSUB2410:SA%=0:GOSUB3800:OPEN"com0:"AS1:IF DA%
=6THEN_COMDTR(,0):CM%=0:RETURN1320ELSE RETURN1320
2150 IF A$=CHR$(13)THEN OP(P)=Y-5+9*(P>3):GOSUB2320
2160 IF A$=CHR$(31)THEN Y=Y+1:IF Y>MX(P)+5-9*(P>3)THEN Y=5-9*(P>3)
2170 IF A$=CHR$(28)THEN PX=1:GOSUB2500
2180 IF A$=CHR$(29)THEN PX=-1:GOSUB2500
2190 IF A$=CHR$(30)THEN Y=Y-1:IF Y<5-9*(P>3)THEN Y=MX(P)+5-9*(P>3)
2200 LOCATE X,Y:PRINT CUR$(1)
2210 GOTO2120
2250 FOR I=0TO6:FOR J=0TO MX(I)
2260 LOCATE CL(I),5+J-9*(I>3):PRINT PR$(J,I)
2270 NEXT J:NEXT I
2280 RETURN
2320 IF OP(0)=1AND P=0THEN GOSUB2610
2330 IF OP(0)=0AND P=0AND PR$(0,0)="300"THEN GOSUB2670
2340 FOR J=0TO MX(P):LOCATE CL(P),5-9*(P>3)+J:PRINT MA$(J,P):PR$(J,P)=MA$(J,P)
2350 IF OP(P)=J THEN A$=MA$(J.P):GOSUB2560:PR$(J.P)=B$:LOCATE CL(P),5-9*(P>3)+J:
```

```
PRINT B$
2360 NEXT J
2370 RETURN
2410 BA=VAL(MA$(OP(0),0)):OP$=MA$(OP(3),3):IF OP$="P"THEN OP$="E"ELSE IF OP$="I"
THEN OP$="O"
2420 OP$=MA$(OP(5),5)+OP$+RIGHT$(STR$(OP(6)+1),1)
2430 IF OP(4)=1THEN OP$=OP$+"N"
2440 IF OP(0)=1THEN DA%=6ELSE IF OP(1)=0THEN IF OP(2)=0THEN DA%=0ELSE DA%=1ELSE
IF OP(2)=0THEN DA%=4ELSE DA%=5
2450 CALL COMINI(OP$,BA,BA):OUT(&H8C),DA%
2460 RETURN
2500 P=P+PX:P=P+7*(P>6)-7*(P<0):IF OP(0)=1THEN P=P-2*(P=1)+2*(P=2):IF P=4THEN P=
P+PX
2510 IF P>3THEN Y=14ELSE Y=5
2520 X=CL(P)-2:RETURN
2560 B$="":FOR SA%=1TO LEN(A$):B$=B$+CHR$(96+ASC(MID$(A$,SA%,1))):NEXT:RETURN
2610 OP(1)=1:OP(2)=0:A$="CCITT":GOSUB2560:PR$(1,1)=B$:LOCATECL(1),6:PRINTB$:B$="
     ":PR$(0,1)=B$:LOCATECL(1),5:PRINTB$:A$="Half-Dupl":GOSUB2560:PR$(0,2)=B$:LOC
ATECL(2),5:PRINT B$:B$="          ":PR$(1,2)=B$:LOCATECL(2),6:PRINTB$
2620 OP(4)=1:A$="No":GOSUB2560:PR$(1,4)=B$:LOCATE CL(4),15:PRINT B$:B$="  ":PR$(
0,4)=B$:LOCATE CL(4),14:PRINT B$
2630 RETURN
2670 B$="Bell":PR$(0,1)=B$:LOCATE CL(1),5:PRINT B$:A$="Originate":GOSUB2560:PR$(
0,2)=B$:LOCATE CL(2),5:PRINT B$:B$="Answer":PR$(1,2)=B$:LOCATE CL(2),6:PRINT B$
2680 OP(4)=1:B$="Si":PR$(0,4)=B$:LOCATE CL(4),14:PRINT B$
2690 RETURN
2730 SOUND0,70:SOUND12,5:SOUND8,63:SOUND13,&B00001000:SOUND7,&B10111110:RETURN
2770 CLS
2780 SN$="":PRINT:PRINT"Graba contenido":PRINT:INPUT"Graba (S/N):";SN$:IF SN$="S
"OR SN$="s"THEN2790ELSE IF SN$="N"OR SN$="n"THEN FR%=1:RETURN ELSE2770
2790 EF%=1:INPUT"Nombre archivo:";AR$
2800 BEEP:OPEN AR$FOR OUTPUT AS2
2810 A$=STRING$(255,32)
2820 A$=USR4(A$)
2830 IF LEN(A$)=0THEN2880
2840 PRINT#2,A$;
2850 IF ER%THEN ER%=0:CLOSE2:RETURN
2860 IF FG THEN2880
2870 GOTO2820
2880 CLOSE2
2890 GOSUB2990
2900 RETURN
2940 SA%=1:POKE GSI,0:POKE CSI,1
2950 FR%=0:RETURN
2990 BEEP:POKE PTR,0:POKE PTR+1,0:POKE PTR+2,0:POKE PTR+3,0:POKE CSI,1:SA%=0:KEY
2,STR$(&H7FFF):FR=0
3000 RETURN
3050 IF INKEY$<>""THEN3050
3060 SN$="":CLS:PRINT"Va a salir de la base de datos":PRINT
3070 PRINT:INPUT"Confirma (S/N):";SN$
3080 IF SN$="n"OR SN$="N"THEN FR%=0:RETURN
3090 IF SN$<>"s"AND SN$<>"S"THEN3060
3100 CLOSE:KEYOFF:DEFUSR=&H3E:A=USR(0):KEYON:BEEP:ON ERROR GOTO0
3110 IF FR<>0THEN GOSUB2780
3120 RUN
3160 DATA1,0,2,1,1,13,1,0,23,2,2,37,1,1,9,3,3,20,2,0,29
3170 DATA300,1200,Bell,CCITT,Originate,Answer,P,I,N,Si,No,5,6,7,8,1,1.5,2
3180 DATA" Tron"," ",Graba," Cls",Param,"Eco:No","LF:No",Status,[-],[-]
3220 ON EF%GOTO3270,3340
3230 RESUME NEXT
3270 IF ERR<>70AND ERR<>19THEN PRINT"Error. Intente Nuevamente":FOR I%=1TO1000:N
EXT I%:CLOSE2:ER%=1:RESUME NEXT
3280 IF ERR=19THEN RESUME NEXT
3290 PRINT"Coloque el disco en la unidad":PRINT"Pulse una tecla para continuar"
3300 A$=INKEY$:IF A$=""THEN3300ELSE RESUME
3340 IFERR=19ORERR=50ORERR=59ORERR=54THEN RESUME NEXT ELSE RESUME NEXT
3380 CLS:LOCATE,,0:CLOSE:PRINT"OPCIONES:"
3390 FOR I=1TO9:KEY(I)OFF:NEXT
```

```
3400 PRINT:PRINT:PRINT"1 - Grabar contenido de memoria"
3410 PRINT"2 - Limpiar memoria"
3420 PRINT"3 - Visualizar memoria"
3430 PRINT"4 - Enviar archivo"
3440 PRINT"5 - Salir de la base de datos"
3450 PRINT:PRINT TAB(4);:PRINT"Ingrese su elección (1-5)"
3460 A$=INKEY$:IF A$=""THEN3460ELSE OP=VAL(A$)
3470 IF A$=CHR$(27)THEN FR%=0:GOTO3510
3480 IF FR=0THEN IF OP<4THEN BEEP:GOTO3460
3490 IF OP<1OR OP>5THEN3460
3500 ON OP GOSUB2770,2990,4180,3550,3050
3510 IF FR%THEN FR%=0:GOTO3380ELSE EF%=2:CLS:OK%=0:FOR I=1TO8-(PEEK(&HFAFC)<>0):
KEY(I)ON:NEXT I:LOCATE,,1:GOTO1280
3550 CLS:SN$=""
3560 PRINT:PRINT"Enviar archivo"
3570 EF%=1:IF PEEK(&HFDEF)<>&HC9THEN FILES:PRINT
3580 OPEN"com0:"AS1
3590 IF(INP(&H82)AND1)=0AND OP(0)=1THEN PRINT"Hay portadora en línea":FOR I=1TO1
000:NEXTI:FR%=1:RETURN ELSE CALL COMDTR(,1):CM%=1
3600 INPUT"Nombre archivo:";AR$
3610 IF AR$=""THEN FR%=1:RETURN
3611 CLS
3612 PRINT "FORMA DE ENVIO":PRINT
3613 PRINT "1) Envio normal"
3614 PRINT "2) Envio a ACAMATICA"
3615 PRINT:PRINT TAB(4);:PRINT"Ingrese su opcion (1-2)"
3616 A$=INKEY$:IF A$="" THEN 3616 ELSE OP=VAL(A$)
3617 ON OP GOTO 3620,5000
3620 OPEN AR$FOR INPUT AS2
3630 IF OP(0)AND PEEK(&HFDEF)<>&HC9THEN LF=LOF(2)ELSE LF=-1
3640 IF ER%THEN ER%=0:CLOSE2:RETURN
3650 IF EOF(2)THEN3750
3660 IF LF=-1THEN3680
3670 IF OP(0)THEN IF LF>100THEN A$=INPUT$(100,2):LF=LF-100:GOTO3690
3680 A$=INPUT$(1,2)
3690 EF%=2
3700 PRINT#1,A$;
3710 IF ER%THEN ER%=0:CLOSE2:RETURN
3720 IF LOC(1)THEN GOSUB1570
3730 IF INKEY$=CHR$(27)THEN3750
3740 EF%=1:GOTO3650
3750 CLOSE
3760 RETURN
3800 SCREEN0:WIDTHWI
3810 LOCATE0,0,0:PRINT TAB(11-15*(WI=80));"MODEM TALENT MSX        v.3.2":PRINTTA
B(12-15*(WI=80));"BANCO DE DATOS":FOR I=1TO WI:PRINT"-";:NEXT
3820 LOCATE,,1
3830 ON KEY GOSUB1640,1690,1730,1780,1900,1820,1840,3880,4110:FOR I%=1TO8-(PEEK(
&HFAFC)<>0):KEY(I%)ON:NEXT:KEYON
3840 RETURN
3880 KEYOFF
3890 FOR II=1TO9:KEY(II)OFF:NEXT
3900 XX=POS(0):YY=CSRLIN
3910 AA$=PR$(OP(1),1):GOSUB4030:CC$=BB$
3920 AA$=PR$(OP(2),2):GOSUB4030
3930 LOCATE0,23:PRINTOP$;BA;CC$;" ";BB$;" ";
3940 IF OP(0)=1THEN IF(INP(&H82)AND1)=0THEN PRINT"Port:Remoto";ELSE IF CM%THEN P
RINT"Port:Local";ELSE IF CM%=0THEN PRINT"Port:No";
3950 IF INKEY$=""THEN3950
3960 LOCATE XX,YY
3970 KEYON
3980 FOR II=1TO8-(PEEK(&HFAFC)<>0):KEY(II)ON:NEXT
3990 RETURN
4030 BB$=""
4040 FOR II=1TO LEN(AA$)
4050 BB$=BB$+CHR$(ASC(MID$(AA$,II,1))-96)
4060 NEXT II
4070 RETURN
```

```
4110 IF WI=40THEN WI=80ELSE WI=40
4120 WIDTH WI
4130 GOTO3800
4180 KEY(1)ON:KEY2,STR$(&H7FFF)
4190 POKE PTR+2,0:POKE PTR+3,0
4200 P1=PEEK(PTR):P2=PEEK(PTR+1)
4210 CLS:PRINT"Visualizar memoria:":PRINT"  Pulse <RETURN> para comenzar":PRINT:
PRINT
4220 A$=INKEY$:IF A$=CHR$(27)THEN4310
4230 IF A$<>CHR$(13)THEN4220
4240 A$=STRING$(WI," ")
4250 A$=USR4(A$)
4260 IF LEN(A$)=0OR INKEY$=CHR$(27)THEN4310
4270 PRINT A$;
4280 A$=USR5(A$)
4290 KEY2,STR$(&H7FFF-PEEK(PTR+2)-PEEK(PTR+3)*256)
4300 GOTO4240
4310 POKE PTR+2,0:POKE PTR+3,0
4320 POKE PTR,P1:POKE PTR+1,P2
4330 IF P2=&H80THEN KEY2,"0000"
4340 PRINT:PRINT:PRINT"Pulse <RETURN> para volver"
4350 IF INKEY$<>CHR$(13)THEN4350
4360 FR%=0:RETURN
4370 END
4998 '
4999 '
5000 CLS:EF%=1:OPEN AR$ FOR INPUT AS 2
5010 IF ER% THEN ER%=0:CLOSE2:RETURN
5030 LINE INPUT#2,A$:IF LEN(A$)>38 THEN A$=LEFT$(A$,38):'INPUT$(38,2)
5040 EF%=2
5050 PRINT #1,A$
5060 IF LOC(1) THEN PRINT:GOSUB 1570
5080 IF EOF(2)=0 THEN GOSUB 5200:GOTO 5030
5100 RETURN
5200 IF LOC(1) THEN GOSUB 1570
5204 PRINT:PRINT"--------------------------------------------------------------
5205 PRINT "Espere la reacción de ACAMATICA":PRINT "<RETURN>: para continuar":PR
INT "<ESC>: para interrumpir envio"
5206 PRINT"------------------------------------------------------
5210 A$=INKEY$
5220 IF LOC(1) THEN GOSUB 1570
5225 IF A$=CHR$(27) THEN CLOSE#2:RETURN
5230 IF A$=CHR$(13) THEN PRINT:RETURN
5240 GOTO 5210
```

POSIBILIDADES DEL PROCESADOR DE TEXTOS

(5ª parte)

Les mostramos como podemos escribir nuestros programas en el MSX WRITE.

En informática denominamos "edición" al proceso por el cual revisamos el listado de un texto creado previamente, con la posibilidad de corregir, suprimir o insertar partes en dicho texto.

En las microcomputadoras existen dos tipos de editores: a) los de LINEA y b) los de PANTALLA. Los primeros actúan línea por línea, de modo tal que si queremos modificar el contenido de una de ellas en particular, hay que indicarlo expresamente por EDIT número de línea. El BASIC de la norma MSX posee un excelente editor del segundo tipo, en el cual, previo al listado de una o más líneas, pueden hacerse las modificaciones mucho más fácilmente.

LA EDICION CON MSX-WRITE

En programas de reducida extensión, el editor que trae el equipo es más que suficiente para resolver cualquier problema, pero ... ¿Qué sucede con los muy largos? A medida que el usuario toma "altura de vuelo" y empieza a confeccionar programas cada vez más completos para su propio uso, comienzan las dificultades.

Tener que buscar por pantalla, tras un LIST, un comando en particular para modificar su sintaxis, o una variable para cambiarle el nombre, puede llegar a representar ... un dolor de cabeza.

Es en estos casos que recomendamos emplear el MSX-Write de TELEMATICA S.A.. Este procesador de textos nos permite cargar un programa en BASIC que haya sido almacenado en ASCII (SAVE "PROGRAM.BAS",A) y modificarlo a nuestro gusto con todas las herramientas de que dispone.

Finalmente, si solemos trabajar con el M-BASIC para poder compilar posteriormente, el procesador de textos nos permite almacenar el programa - una vez editado - en formato ASCII, con lo cual es asequible al compilador.

LA CREACION DEL PROGRAMA

Otra de las posibilidades es crear directamente el programa con el MSX-Write, almacenarlo en disco o en casete, y luego - desde el BASIC - cargarlo normalmente y ejecutarlo. Esto es posible porque el MSX-Write almacena directamente (sin necesidad de especificárselo) en código ASCII, y por ello en formato equivalente al SAVE "...",A.

Obviamente es un requisito para esta modalidad operativa tener previamente el programa "predigerido", ya que no está disponible la facilidad del intérprete BASIC de ir probando -parcialmente- el programa para ver si realmente hace lo que deseamos...

OTRA POSIBILIDAD

Incorporar en el programa rutinas o subrutinas que tengamos previamente almacenadas en formato ASCII, a las cuales podemos acceder con la opción BLOQUE - CARGAR, sin necesidad de que la numeración de líneas sea la correcta es otra posibilidad que se nos brinda con el procesador de textos.

La ventaja que esto otorga es la de no perder líneas del programa al cual le agregamos una subrutina cuando hay superposición (coincidencia) en el número de líneas, cosa que suele ocurrir al usar el MERGE en BASIC.

Gustavo O. Delfino

CONCURSO MENSUAL

PREMIO

MSX-TEST



Un soft a elección entre IDEA BASE, IDEA TEXT, o BASIC TUTOR.

CONCURSO MENSUAL

Auspiciado por TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Para participar en este certamen deben señalar cual es la información correcta que presenta cada ítem. Junto con las respuestas deben remitir los datos en el correspondiente cupón. Los premios se entregarán en la administración de la editorial. Los que no puedan concurrir pueden solicitar el envio pagando el franqueo contrareembolso. Los premios podrán ser reclamados dentro de los 120 días después de haber sido anunciados.

Ganador del MSX-TEST Nº 6

Jorge Marcelo Linares, Brandsen (Buenos Aires)

Respuestas correctas

1- De información perfecta con movimientos del azar. 2- OPEN. 3- 1200 a 9600 baudios. 4- Bloque de Control de Archivo. 5- De estrategia.

MSX-TEST N.10 - CIERRE 1 DE DICIEMBRE DE 1988

1.¿Cuál es la versión de MINILAN que acaba de lanzar TELEMATICA?
a) II.
b) III.
c) IV.

2.¿Qué hace el comando WIDTH?
a) Compara dos palabras.
b) Fija el ancho de columnas.
c) Hace un scroll de pantalla.

3.¿Qué significa CP/M?
a) Programa de control para microcomputadoras.
b) Mantenimiento de computadoras personales.
c) Lenguaje C para microcomputadoras.

4.180 es un programa que simula...
a) una carrera de autos.
b) una carrera de motos.
c) el juego de los dardos.

5.ARGOS es un software...
a) utilitario.
b) de juegos.
c) educativo.

Nombre y apellido____________________

Dirección:____________________

Documento:__________ Edad:_____ Máquina:__________

Qué es lo que más me gusta de la revista:____________________

Que le agregaría:____________________

Que es lo que no me gusta:____________________

INDIANA JONES Y EL TEMPLO DE LA PERDICION

Indiana Jones, Indy para los amigos, es un personaje bien conocido por el gran suceso que tuvieron las películas de Spielberg en todo el mundo.
Harrison Ford personifica en el cine a este aventurero y sirve como modelo para la presentación del juego.
La historia comienza cuando nuestro héroe llega a una población que está sufriendo varias desgracias: malas cosechas, niños raptados, etcétera. El origen de todos estos males fue la desaparición de la piedra sagrada, que protegía la ciudad.
Indy es un importante arqueólogo y recuerda la leyenda de las piedras sankara. Eran cinco piedras que concedían poderes mágicos a sus poseedores. Hace mucho tiempo esas piedras se desperdigaron provocando la desaparición de la tribu thuggee que las adoraba.



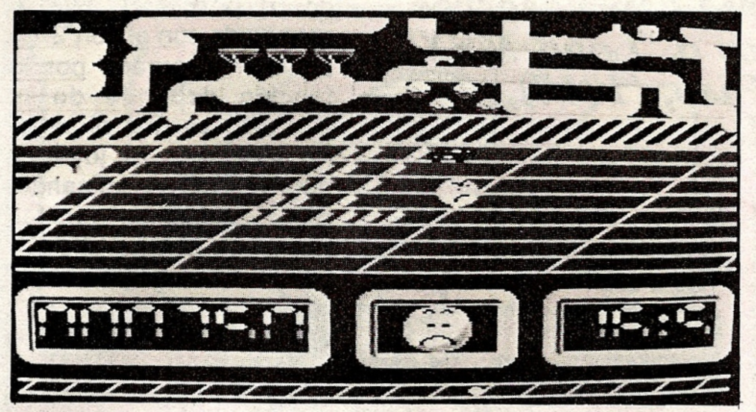
Las piedras, según la leyenda, eran de un tamaño inferior a los veinte centímetros y tenían forma de un dedal ovalado. También se suponía que al juntarse cada una con su compañera brillaba en su interior una luz, como si la cercanía aumentara su poder.
Luego de recabar datos, Indy se dirige a la montaña de donde los ancianos le habían dicho que provenía el mal. Cuando se acerca a la montaña escucha unos alaridos que lo ponen alerta. Los gritos proveinen de unos bultos que no consigue distinguir por la distancia que lo separa de ellos. Comprueba horrorizado que esos bultos eran jaulas donde hay niños encerrados, cuya apariencia externa indica malos tratos y desnutrición.
Cuando se acerca a una de las jaulas para liberar a uno de los pequeño ve cómo una sombra se acerca peligrosamente a él. Sin dudar un momento desenfunda el látigo y dándose vuelta lo lanza contra la sombra que cae desvanecida a unos pasos de él.
La vestimenta utilizada por aquel personaje le recuerda la leyenda de los guerreros thuggees; surge la posibilidad entonces de que la misma sea realidad.
Con un certero latigazo abre la jaula en donde estaba encerrado el niño, luego vuelve a desmayar al guardia y corre montaña arriba.



Siempre armado con su látigo Indy rescata a otros niños y hace frente a los guardias y a las serpientes que aparecen.
Finalmente un niño le cuenta lo que sucedia. Mola Ram había logrado reunir tres de las cinco piedras sankara, y con ellas reunió nuevamente la secta thuggee. Ahora utilizaba a los niños para encontrar las dos restantes. A los que intentaban escapar los enjaulaba para que sirvieran de ejemplo atemorizador.
Este niño, agradecido con Indy, le indica la situación exacta de la entrada a la mina donde los niños buscaban las piedras sankara.
Indy se dirige a la entrada de la mina y se introduce en ella en una vagoneta. Los thuggees lo atacan tratando de hacerlo descarrilar. Para su desgracia la vagoneta tampoco tiene frenos. Tras algunos contactos con los thuggees llega al final del recorrido, donde encuentra una puerta que lleva al templo. Para concluir la aventura tiene que cruzar un río de lava y llegar al altar de los sacrificios, donde se encuentran las piedras sànkara. Tiene que emplear una vez más su látigo para aprovechar una saliente en la roca y salvar el obstáculo.
Luego de tomar el talismán sale del templo perseguido por los guerreros y consigue retornar la piedra a su lugar original. La paz, la felicidad y los niños retornan a la ciudad.
El juego es reflejo de la película. Indy con su látigo debe luchar con los guerreros thuggees y con Mola Ram. Los niños están cautivos y él es el encargado de rescatarlos. (REAL TIME)

ALE HOP

Ale es un muñequito en forma de pelota que tiene que transitar por un laberinto en la menor cantidad de tiempo posible.
Debe enfrentarse con diversas trampas que lo harán saltar, retroceder y darse toda clase de golpes.
Obviamente hay que tratar de evitarlos. Para llegar al final de la etapa, hay que pasar por encima de unos fotones. Como recompensa por esta acción tendremos segundos adicionales que se transformarán en puntos cuando hayamos logrado nuestro objetivo.
Ale comienza su viaje muy feliz pero, a medida que recibe los golpes, vemos que se vuelve cada vez más "caracúlico", y con razón, ya que es difícil evitar el dolor de las caídas.
El juego consiste en una combinación de habilidad y velocidad. Los gráficos abarcan casi toda la pantalla y se componen de una parte quieta y otra que se mueve por delante o detrás de ella. Esto se ve sobre todo en el cambio de etapa. Allí Ale se pone el casco y se sube a una nave que lo conduce al comienzo de la siguiente etapa. Para los programadores hacemos notar este efecto, ya que con el simple desplazamiento de estrellas se consigue un buen efecto.
Atención: para comenzar la segunda etapa hay que pul-



sar la barra espaciadora, si no el viaje sigue indefinidamente.

En la segunda etapa, nuevos tropiezos esperan a nuestro amigo: corrientes eléctricas, cintas sinfín, etcétera. Las caras de susto son entonces comprensibles.

Si a todo lo dicho le sumamos una pegadiza melodía, podemos concluir que se trata de un juego bien hecho, que toma ideas de otros programas y las perfecciona. Lo recomendamos a los que quieren poner a prueba sus reflejos. (REAL TIME)

NEWLETTERS SYSTEM

Este es un utilitario que nos permite disponer de ¡cuarenta y tres! tipos diferentes de letras.

Podremos imprimir un archivo en cualquiera de esos tipos. Para ello es necesario seguir los pasos 2, 3 y 4, que



son los seteos de letras, impresora y archivos, para recién después efectuar la impresión.

El sistema también cuenta con una prueba de impresión y con la opción de ver el tipo de letra por pantalla o no. En algunos casos es conveniente, en otros (letras para títulos) se hace imposible leer en la pantalla lo que está escrito.

Muchas letras ofrecidas por este sistema resultarán conocidas a los usuarios fanáticos de los videogames, ya que son las mismas de los juegos. Allí están, por ejemplo, Hunter, Nomad, Rocket, Ghost y Beret, entre otras.

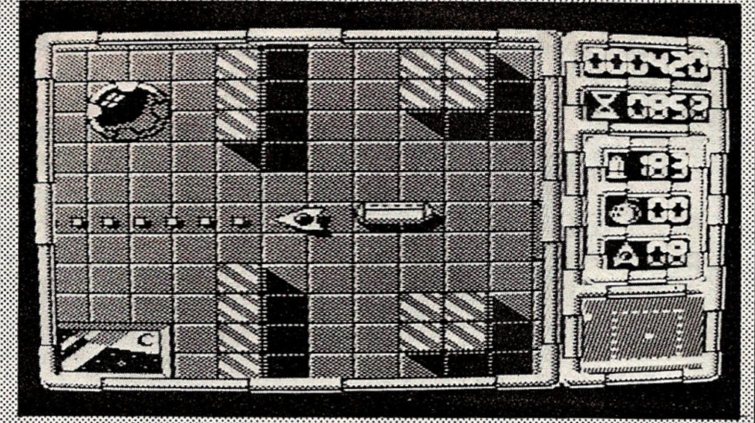
También tenemos tipos de letras más tradicionales como itálica o de computación.

Se pueden conseguir efectos especiales con letras sombreadas, locas, de "comics", etcétera.

La impresión de archivos se puede hacer por una impresora Epson compatible, Seikosha GP-550 o Gemini.

Las alternativas que ofrece este utilitario son lo suficientemente interesantes como para tenerlo entre nuestros programas. (REAL TIME)

ASTEROIDS



Una vez más debemos ponernos al mando de una nave espacial. La misión es defender nuestra base de los invasores que toman las más diversas formas. Para conseguir el objetivo tendremos la ayuda de la tecnología de la nave, en especial el radar que nos indicará la posición de los enemigos.

Como el comportamiento de la nave es inercial, la mejor forma de controlarla es mantener siempre una velocidad moderada y maniobrar con pequeños toques de aceleración. Al avanzar en una dirección conviene hacerlo de espaldas, para que se pueda frenar rápidamente si aparece algún enemigo.

Cuando disparamos a un invasor no lo destruimos completamente, sino que lo fragmentamos en trozos más pequeños. Por esa razón es conveniente disparar ráfagas en forma de abanico, para conseguir destruir también los nuevos entes.

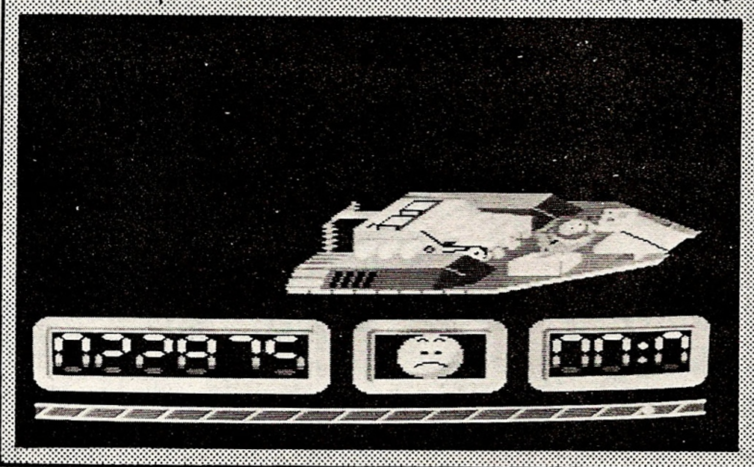
La bomba es un arma muy útil que se debe utilizar con cautela, ya que son limitadas.

Para acabar con todos los asteroides de una etapa la mejor táctica es trazarse una ruta que recorra la pantalla liquidando a todos los que veamos. Recién después recorremos el laberinto a mayor velocidad, pues el peligro será menor.

De vez en cuando nos cruzaremos con pelotas que nos darán puntos o tiempo extra si las recogemos.

Con la opción "A" del menú se evita volver a comenzar un partido si perdemos en una etapa avanzada.

Un último truco: si mientras se está jugando se aprietan las teclas E, L, I y S, seremos inmunes a los diferentes enemigos que enfrentamos. No nos queda más que desearles suerte en la misión. Nuestro futuro está en sus manos. (REAL TIME)



Para comunicarse con nosotros deben escribirnos a 'Load MSX", Paraná 720 ,5º Piso, (1017), Capital Federal.

ADAPTADOR

Quisiera felicitarlos por la calidad de los trabajos presentados en la revista y aprovechar para hacerles unas consultas. ¿Cómo tendría que hacer si quisiera tener la unidad de disco y el modem encendidos al mismo tiempo?
¿Existen para MSX juegos de estrategia en que uno es un general y tiene que planificar una guerra desplegando ejércitos?
Un último pedido: ¿podrían publicar mi dirección para poder cartearme con otros usuarios? Desde ya muchas gracias y sigan adelante.

Guillermo A. Di Camillo
Rodríguez Peña 1560
1663 San Miguel
Prov. de Buenos Aires

LOAD MSX

El problema del modem y la unidad de disco se soluciona con el adaptador de Talent APT-950.
Con respecto a la consulta sobre el juego, lamentablemente no existen juegos de este tipo disponibles para MSX.
Publicamos tu dirección para alentar el intercambio de programas e información entre los usuarios.

AUTO

Les escribo para que me amplíen el uso de la sentencia AUTO y NEW, ¿para qué sirven?

Carlos María Buitrago
Mar del Plata

LOAD MSX

AUTO es un comando que genera los números de línea automáticamente. La forma de usarlo es AUTO n,m donde n es el primer número de línea que queremos generar, mientras que los siguientes números están separados por el incremento m. Por ejemplo, AUTO 100,10 generará números de línea 100, 110, 120, etcétera.
NEW es un comando que borra el programa BASIC residente en la memoria de la máquina. Puede ser en una línea de programa, pero, como borra éste de la memoria tan pronto como es ejecutado, no existen muchas aplicaciones para él en este modo. Los programadores expertos lo pueden usar junto a un condicional para proteger sus programas.

PLOTTER

Mi pregunta es muy corta, pero desearía que me la aclaren: ¿qué es un plotter?

Daniel Sella
Santa Rosa - LA PAMPA

LOAD MSX

El plotter es un trazador de gráficos, es decir un dispositivo de salida para traducir la información de la computadora en forma gráfica sobre papel o un soporte similar.
Existen plotters para todas las necesidades de tamaños, exactitud, velocidad y hasta color.



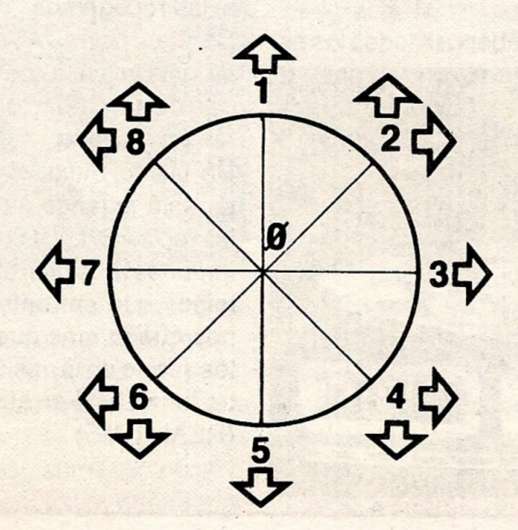
JOYSTICKS

Me gustaría saber cómo puedo hacer para manejar o utilizar el joysticks en mis programas. ¿Hay alguna forma sencilla de hacerlo? Por lo demás aprovecho para felicitarlos por la revista y enviarles un gran saludo.

Marcos Fischer
Victoria
Prov. de Buenos Aires

LOAD MSX

El comando usado para saber en qué dirección está manejando o simulando el joystick es STICK(n). n tiene un valor entre 0 y 2; 0 indica que la instrucción leerá las teclas del cursor, como si fueran un joystick; n=1 hace referencia al joystick conectado al port 1 y n=2 al conectado con el port 2.
El valor retornado por la función depende de la dirección en que el usuario desplaza el joystick que se está examinando. Si lo que utiliza son las teclas de cursor, el valor retornado indica la combinación de teclas pulsadas a la vez. En la figura se muestra los valores retornados para la diferentes direcciones.
Si no se pulsa ninguna tecla al evaluar la función, o si el joystick no se mueve, la función da el valor 0.



BASIC vs ASSEMBLER

He hecho varios programas en BASIC, y me gustaría aprender lenguaje ASSEMBLER. ¿Qué ventajas pueder representar el uso de este lenguaje?

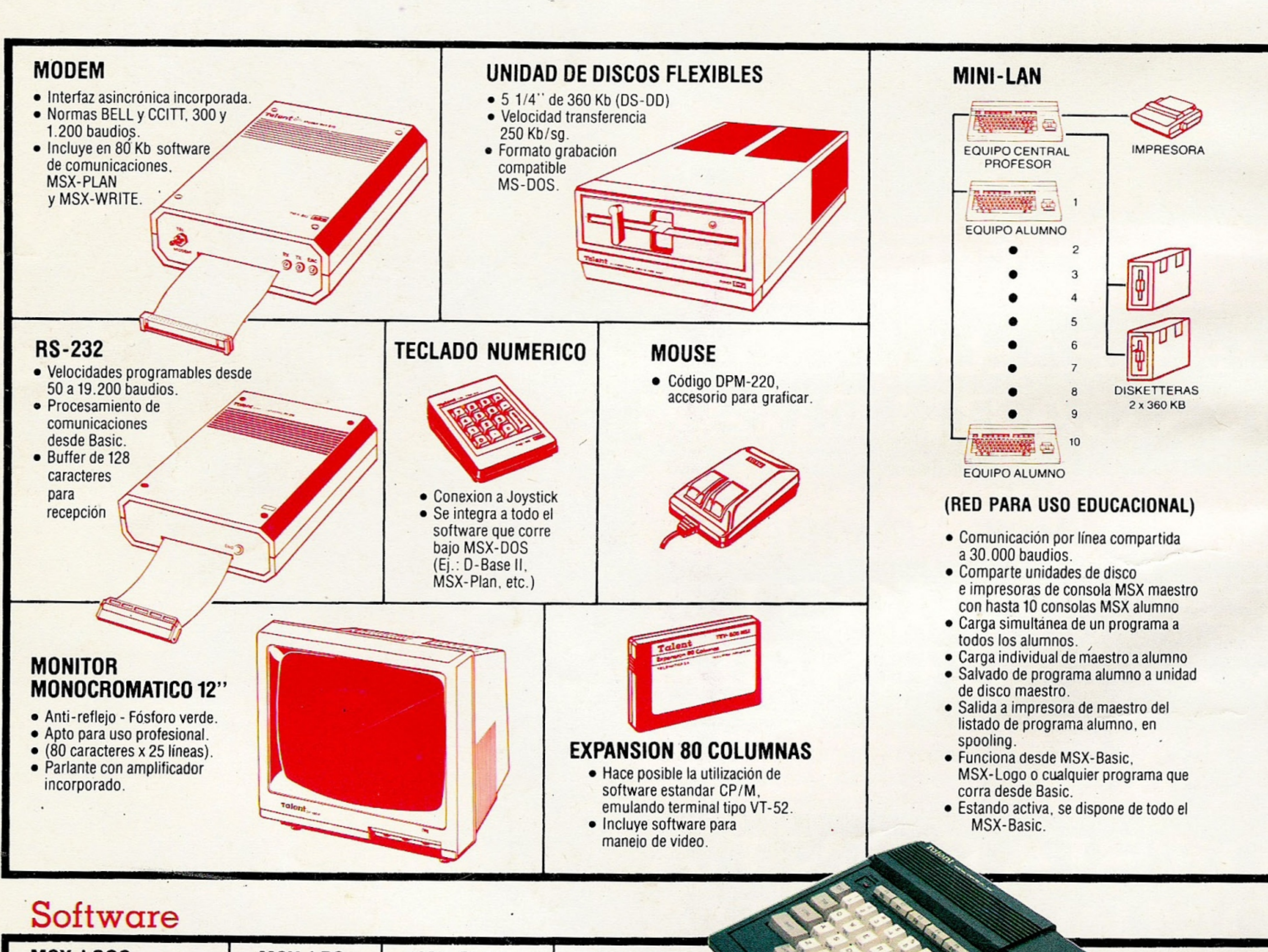
Martín Cabezas
Bahía Blanca

LOAD MSX

Casi todas las computadoras están equipadas con BASIC. La razón de esto es que es un lenguaje relativamente sencillo. Especialmente el MSX BASIC sobresale con su gran variedad de comandos.
Sin embargo, el ASSEMBLER cuenta con dos ventajas respecto al BASIC:la velocidad de ejecución de los programas y el ahorro considerable de memoria.
A la hora de hacer los programas hay que elegir qué lenguaje conviene utilizar de acuerdo a la características del mismo.


SEPTIEMBRE 1988
COMPUTACION
K64
PARA TODOS
★DESDE EE.UU.★
SUPLEMENTO
BYTE
LA SOCIEDAD
CABLEADA
UTILITARIOS
PARA SPECTRUM
Y COMMODORE
₳25.000
EN PREMIOS
MSX2 TPC-310
SPECTRUM - CZ - TK - DREAN COMMODORE - TI - MSX - ATARI y PC

Encienda una computadora Talent MSX y sus periféricos.



Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de cálculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.

MSX DPC-200

Talent
Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.